A Fuzzy Framework for Human Hand Motion Recognition

Zhaojie Ju
School of Creative Technologies
University of Portsmouth

A thesis submitted for the degree of
Doctor of Philosophy
June 2010
Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.
This thesis is dedicated to my parents, and my wife, Deman Zhao, with love.
Acknowledgements

Foremost, I wish to express my sincere gratitude to my supervisor, Dr. Honghai Liu, for his unflinching support, scientific guidance and warm encouragement. His joy and enthusiasm for his research is contagious and motivational and his truly science intuition has made him as a constant oasis of ideas and passions in science.

Of equal importance are my thanks to Prof. Youlun Xiong and Dr. David J. Brown for their continuous support, constructive comments and helpful suggestions on my research and career development.

I would also like to thank my colleagues at Institute of Industrial Research and Intelligent Systems & Biomedical Robotics Group. Special thanks go to Dr. Jiangtao Cao and Mrs. Xiaofei Ji for their consistent encouragement and sharing their knowledge and expertise with me. Ruiying Wang is acknowledged for her outstanding knowledge in simulation and hardware test, and for always being helpful and sharing her time. I appreciate the participations during my experiments from Mr. Xin Wen, Mr. Geoffrey Samuel, Mr Ping Liu and Mrs Justine Chaves. I am also happy to work with Mr. Mehdi Khoury, Dr. Chee Seng Chan, Dr. Ian Morgan, Mr. Edward Smart, Dr. Josh Fahimi, Mr. Yuhui Shao during my years as PhD student.

I am grateful to my family for their unconditional love support throughout all my studies, without which this work would not be possible.
Abstract

Unconstrained human hand motions consisting of grasp motion and in-hand manipulation lead to a fundamental challenge that many recognition algorithms have to face, in both theoretical and practical development, mainly due to the complexity and dexterity of the human hand. The main contribution of this thesis is a novel fuzzy framework of three proposed recognition algorithms. This consists of extended Time Clustering (TC), Fuzzy Gaussian Mixture Model (FGMM) and Fuzzy Empirical Copula (FEC), using numerical values, Gaussian pattern and data dependency structure respectively in the context of optimal real-time human hand motion recognition.

First of all, a fuzzy time-modeling approach, TC, is proposed based on fuzzy clustering and Takagi-Sugeno modeling with a numerical value as output. The extended TC is not only capable of learning repeated motions from the same subject but also can effectively model similar motions from various subjects. The recognition algorithm itself can identify the start point and end point of the testing motion. It is applicable to motion planning directly transferred from the recognition result.

Secondly, FGMM is developed to effectively extract abstract Gaussian patterns to represent components of hand gestures with a fast convergence. The dissimilarity function in fuzzy C-means, which maintains the exponential relationship between membership and distance, is refined for FGMM with a degree of fuzziness in terms of the membership grades. Not only does it possess non-linearity but it also offers the characteristic of computationally inexpensive convergence. It is applicable to applications which have a small model storage space and require a method to generate the desired trajectory.
Thirdly, FEC is proposed by integrating the fuzzy clustering by local approximation of memberships with Empirical Copula (EC). To save the computational cost, fuzzy clustering reduces the required sampling data and maintains the interrelations before data dependence structure estimation takes over. FEC utilizes the dependence structure among the finger joint angles to recognize the motion type. It is capable of effectively recognizing human hand motions for both single subject and multiple subjects with a few training samples. It can be used in applications requiring high recognition rate and no desired trajectory with limited training samples.

All the proposed algorithms have been evaluated on a wide range of scenarios of human hand recognition: a) datasets including 13 grasps and 10 in-hand manipulations; b) single subject and multiple subjects. c) varying training samples. The experimental results have demonstrated that all the proposed methods in the framework outperform Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) in terms of both effectiveness and efficiency criteria.
# Contents

List of Figures ix

List of Tables xiii

1 Introduction 1
   1.1 Background ............................................. 1
   1.2 Problems and Challenges ................................. 3
   1.3 Overview of Approaches and Contributions .......... 4
   1.4 Outline of Thesis ..................................... 5

2 Literature Review 8
   2.1 Introduction ........................................... 8
   2.2 Hand Motion Capture ................................... 12
      2.2.1 Glove based Capturing ............................. 12
      2.2.2 Vision based Capturing ............................ 17
         2.2.2.1 Single Camera ................................ 17
         2.2.2.2 Multiple Cameras ............................. 18
      2.2.3 Electromyogram based Capturing .................. 19
   2.3 Hand Modeling .......................................... 20
      2.3.1 Kinematic Hand Model .............................. 21
      2.3.2 Static Hand Model ................................ 23
      2.3.3 Dynamic Hand Model .............................. 25
   2.4 Hand Motion Recognition ............................... 27
      2.4.1 Hidden Markov Models .............................. 28
      2.4.2 Finite State Machine .............................. 31
2.4.3 Connectionist Approach ........................................ 32
2.4.4 Other Recognition Methods ................................... 34
2.5 Summary ............................................................ 36

3 Time Clustering .......................................................... 38
  3.1 Introduction .......................................................... 38
  3.2 Takagi-Sugino Fuzzy Modeling .................................... 39
  3.3 Time Clustering Recognition ....................................... 40
    3.3.1 Motion Model Construction ................................... 40
    3.3.2 Motion Recognition ............................................ 41
  3.4 Validation .......................................................... 42
  3.5 Summary ............................................................ 46

4 Fuzzy Gaussian Mixture Models ........................................ 47
  4.1 Introduction .......................................................... 47
  4.2 Generalized Gaussian Mixture Models ......................... 49
    4.2.1 Conventional Gaussian Mixture Models .................... 49
    4.2.2 Generalized Gaussian Models ............................... 51
    4.2.3 Modified Expectation Maximisation Algorithm ............ 54
  4.3 Fuzzy Gaussian Mixture Models .................................. 55
    4.3.1 Fuzzy C-means Clustering ................................... 56
    4.3.2 Probability based Fuzzy Gaussian Mixture Models ........ 58
    4.3.3 Distance based Fuzzy Gaussian Mixture Models .......... 59
    4.3.4 Comparison of Probability/Distance based Fuzzy Gaussian Mixture Models ........................................... 61
    4.3.5 Mathematical Proof .......................................... 62
  4.4 Validation .......................................................... 63
    4.4.1 Gaussian based Datasets ...................................... 64
      4.4.1.1 Five Components Gaussian ............................... 65
      4.4.1.2 Various Components Gaussian .......................... 66
    4.4.2 Structure Analysis of Characters ......................... 70
  4.5 Fuzzy Gaussian Mixture Models based Recognition ............ 76
  4.6 Summary ............................................................ 78
## Fuzzy Empirical Copula

5.1 Introduction .................................................. 79
5.2 Dependence Structure Estimation via Empirical Copula .......... 82
  5.2.1 Copula .................................................. 83
  5.2.2 Empirical Copula and Dependence Estimation .......... 84
5.3 Fuzzy clustering by Local Approximation of Memberships .... 85
  5.3.1 Initialization ........................................... 86
  5.3.2 Approximation ........................................... 87
  5.3.3 Assignment .............................................. 88
5.4 Fuzzy Empirical Copula ........................................ 89
5.5 Validation ...................................................... 93
  5.5.1 Testing Datasets .......................................... 93
  5.5.2 Dependence Structure Estimation via Empirical Copula .. 94
  5.5.3 Dependence Structure Estimation via Fuzzy Empirical Copula 94
  5.5.4 Comparison of Flame$^+$ to K-means for EC .............. 96
5.6 Recognizing Human Hand Motion via Fuzzy Empirical Copula ... 100
  5.6.1 One-to-one correlation and motion template ............... 100
  5.6.2 Motion Recognition ..................................... 102
5.7 Summary ....................................................... 102

## Experiments and Evaluation

6.1 Introduction .................................................. 104
6.2 Experiment Set-up ............................................. 105
  6.2.1 Grasp Motions .......................................... 106
  6.2.2 In-hand Manipulation .................................... 106
  6.2.3 Note on Data Analysis .................................... 108
6.3 Experiment 1- Training with One Sample from a Single Subject, Comparing within the Subject ......... 108
6.4 Experiment 2- Training with Six Samples from Multiple Subjects, Comparing Across Multiple Subjects ................. 110
6.5 Experiment 3- Training with Multiple Samples from a Single Subject, Comparing within the Subject ................. 112
## List of Figures

1.1 Learning process from human hand motion to multifingered manipulation ................................................. 2
2.1 Bones of the human hand and forearm. Individual bone names are underlined. The metacarpal, proximal phalanx, and distal phalanx bones exist in each finger of the human hand, while the middle phalanx bones exist in all fingers but the thumb (Albrecht et al., 2003). .................................................. 9
2.2 The main fundamental phases of the learning process for multifingered manipulation .................................. 11
2.3 Hand motion capture of the learning process for multifingered manipulation ........................................... 12
2.4 Data gloves on the market ......................................................................................................................... 15
2.5 Prototype of StrinGlove(Left) and Prototype of StrinGlove(Right) (Kuroda et al., 2004) ............................. 16
2.6 Forearm muscle positions ......................................................................................................................... 19
2.7 Hand modeling of the learning process for multifingered manipulation (Wu et al., 2001) ......................... 21
2.8 Cardboard hand model (Bray et al., 2004b) ............................................................................................ 23
2.9 Handling self-occlusion: The 3D model (left) and its generated contour (right) are shown (Stenger et al., 2001). .................................................. 24
2.10 Different ways to model the surface Dewaele et al. (2003). ................................................................. 25
2.11 The hand model as polygonal surface (top). The hand model and its degrees of freedom (bottom) (Bray et al., 2004b). .................................................. 26
2.12 Hand modeling of the learning process for multifingered manipulation .................................................. 27
2.13 A five states and four observations left-to-right HMM for grasp recognition ........................................ 29
3.1 Model of TC based recognition .................................................. 42
3.2 Recognition result of TC .......................................................... 42
3.3 Selected grasping tasks for TC .................................................. 43
3.4 An example of TC based recognition ......................................... 44
3.5 Recognition rate with different number of training data ............... 45
4.1 Generalized Gaussian Model (A) and projection points on the principal curve axis (B). ............................................................ 53
4.2 Fitting Chinese character letter using conventional GMMs (left) with 7 components and generalized GMMs (right) with 4 components.... 56
4.3 Test data set: Data of five conventional Gaussian mixtures (left) and data of five generalized Gaussian mixtures (right). ............... 64
4.4 Results of conventional GMMs (56 iterations, 0.257 second, left), probability based FGMMs ($a = 0$) (48 iterations, 0.235 second, middle) and distance based FGMMs ($a = 0$) (35 iterations, 0.232 second, right) with 5 components on the dataset of five conventional Gaussian mixtures. 65
4.5 Log-likelihoods during convergences of conventional GMMs, probability based FGMMs ($a = 0$) and distance based FGMMs ($a = 0$) EM algorithms with 5 components on the dataset of five conventional Gaussian mixtures. ............................................................. 66
4.6 Results of generalized GMMs (11 iterations, 2.041 seconds, left), probability based FGMMs ($|a| \geq 0$) (7 iterations, 1.562 seconds, middle) and distance based FGMMs ($|a| \geq 0$) (8 iterations, 1.516 seconds, right) with 5 components on the dataset of five generalized Gaussian mixtures. ............................................................. 67
4.7 Log-likelihoods during convergences of generalized GMMs, probability based FGMMs ($|a| \geq 0$) and distance based FGMMs ($|a| \geq 0$) EM algorithms with 5 components on the dataset of five generalized Gaussian mixtures. ............................................................. 68
4.8 Results of conventional GMMs (1263 iterations, 10.75 seconds, left), probability based FGMMs ($a = 0$) (520 iterations, 6.41 seconds, middle) and distance based FGMMs ($a = 0$) (473 iterations, 4.05 seconds, right) with 9 components on the character ‘R’. .................. 71
LIST OF FIGURES

4.9 Log-likelihoods during convergences of conventional GMMs, probability based FGMMs \((a = 0)\) and distance based FGMMs \((a = 0)\) EM algorithms with 9 components on the character ‘R’. .......................... 72

4.10 Results of generalized GMMs (649 iterations, 133.19 seconds, left), probability based FGMMs \((|a| \geq 0)\) (257 iterations, 55 seconds, middle) and distance based FGMMs \((|a| \geq 0)\) (197 iterations, 40.75 seconds, right) with 7 components on the character ‘R’. .......................... 73

4.11 Log-likelihoods during convergences of generalized GMMs, probability based FGMMs \((|a| \geq 0)\) and distance based FGMMs \((|a| \geq 0)\) EM algorithms with 7 components on the character ‘R’. .......................... 73

4.12 An example of DFGMMs based in-hand manipulation motion learning .......................... 76

4.13 An example of DFGMMs based in-hand manipulation motion recognition .......................... 77

5.1 Clustering random 3D Euclidean positions using FLAME\(^+\). The star points in black are the centres of the clusters (CSO); points labeled with triangles are the outliers; the colour range of NPs represents their membership degrees. .......................... 88

5.2 The interrelations of length, diameter, whole weight and shell weight of abalone dataset .......................... 93

5.3 The result of dependences of 36 correlations of the original abalone dataset .......................... 94

5.4 Result of sampled abalone data using 12-neighbour-FLAME\(^+\) .......................... 95

5.5 Comparison of Spearman’s rhos and Gini’s gammas. Blue lines are the correlations of EC algorithm while red lines are of FEC algorithm .......................... 96

5.6 The relationship between number of nearest neighbours and number of abalone’s CSOs .......................... 97

5.7 Change of overall errors of Spearman’s rhos and Gini’s gammas with the growth of number of nearest neighbours in FLAME\(^+\) .......................... 97

5.8 Change of time costed by FEC with the growth of number of nearest neighbours in FLAME\(^+\) .......................... 98

5.9 Change of time cost when comparing the proposed FEC and K-means based Empirical Copula (KEC) .......................... 99
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.10</td>
<td>Changes of overall errors of Spearman’s rhos and Gini’s gammas when comparing the proposed FEC and KEC</td>
<td>100</td>
</tr>
<tr>
<td>5.11</td>
<td>An example of the motion template</td>
<td>101</td>
</tr>
<tr>
<td>6.1</td>
<td>Positions of the recorded hand joint angles</td>
<td>105</td>
</tr>
<tr>
<td>6.2</td>
<td>Experiment grasps</td>
<td>106</td>
</tr>
<tr>
<td>6.3</td>
<td>Four examples of pencil in-hand manipulation</td>
<td>107</td>
</tr>
<tr>
<td>6.4</td>
<td>Recognition results of different motions from different subjects on grasp dataset with one sample from single subject as training data</td>
<td>109</td>
</tr>
<tr>
<td>6.5</td>
<td>Recognition results of different motions from different subjects on in-hand manipulation dataset with one sample from single subject as training data</td>
<td>109</td>
</tr>
<tr>
<td>6.6</td>
<td>Recognition results of different motions from different subjects on grasp dataset with six samples from multiple subjects as training data</td>
<td>111</td>
</tr>
<tr>
<td>6.7</td>
<td>Recognition results of different motions from different subjects on in-hand manipulation dataset with six samples from multiple subjects as training data</td>
<td>111</td>
</tr>
<tr>
<td>6.8</td>
<td>Recognition results with various training percentages from single subject on grasp dataset</td>
<td>113</td>
</tr>
<tr>
<td>6.9</td>
<td>Recognition results with various training percentages from single subject on in-hand manipulation dataset</td>
<td>113</td>
</tr>
<tr>
<td>6.10</td>
<td>Recognition results with various training percentages from multiple subjects on grasp dataset</td>
<td>115</td>
</tr>
<tr>
<td>6.11</td>
<td>Recognition results with various training percentages from multiple subjects on in-hand manipulation dataset</td>
<td>115</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Data gloves available in the market ........................................ 14
3.1 Recognition results of the methods ...................................... 44
4.1 Results of conventional GMMs (CGMM), probability based FGMMs (PFGMM, $a = 0$) and distance based FGMMs (DFGMM, $a = 0$) with 3-7 components on the dataset of five conventional Gaussian mixtures. 69
4.2 Results of generalized GMMs (GGMMs), probability based FGMMs ($|a| \geq 0$) and distance based FGMMs ($|a| \geq 0$) with 3-7 components on the dataset of five generalized Gaussian mixtures. 70
4.3 Results of all models with $k$ components and $m$ degree of fuzziness on datasets of ‘6’, ‘8’, ‘a’, ‘B’ and ‘R’. ................................. 74
5.1 Number of clusters corresponding to number of nearest neighbors for K-means ................................................................. 98
6.1 10 types of in-hand manipulations ...................................... 107
6.2 Recognition results of FEC, TC, FGMM, GMM and HMM with only one sample from single subject as training data .................. 108
6.3 Recognition results of FEC, TC, FGMM, GMM and HMM with six samples from multiple subjects as training data .................. 110
6.4 Time cost by modeling and recognizing of FEC, TC, FGMM, GMM and HMM ................................................................. 116
Chapter 1

Introduction

1.1 Background

The human hand is capable of fulfilling various everyday-life tasks using the combination of biological mechanisms, sensors and controls. It is attracting more and more research interest to build human-like robotic hands with similar dexterous manipulation, which can be of great help in humanoid robots, industry and healthcare. Recent innovations in motor technology and robotics have achieved impressive results in the hardware of robotic hands such as the Southampton Hand, DLR hands, Robonout hand, Barret hand, Emolution hand and Shadow hand (Bicchi, 2000; Light & Chappell, 2000; Lotti et al., 2005; Wei et al., 2005). Especially, the ACT hand (Matsuoka et al., 2006) has not only the same kinematics but also the similar anatomical structure with the human hand, providing a good start for the new generation of anatomical robotic hands.

However, autonomously controlling multifingered robots is still a challenge, which is related to multidisciplinary research and a wide spectrum of applications in intelligent robotics.

Human-robot skill transfer is of great importance in order to develop advanced multifingered manipulation planning and control systems. Zoliner et al. (2005) and Chella et al. (2004) confirmed that prior knowledge should be introduced in order to achieve fast and reusable learning in behavioural features, and integrate it in the overall knowledge base of a system, in order to fulfill requirements such as reusability, scalability, explainability and software architecture (Jacobsson et al., 2008; Verschure et al., 2003). Bae et al. (2005) also showed that characteristics of human grasping
1.1 Background

could enhance dexterity in Robotic Grasping. Robertsson *et al.* (2007) and Carrozza *et al.* (2006) designed artificial hand systems for dexterous manipulation; they showed that human-like functionality could be achieved even if the structure of the system is not completely biologically inspired. Learning from human hand motions is preferred for human-robot skill transfer in that, unlike teleoperation-related methods, it provides non-contact skill transfer from human motions to robot motions by a paradigm that can endow artifacts with the ability for skill growth and life-long adaptation without detailed programming (Calinon *et al.*, 2007). In principle, not only does it provide a natural, user-friendly means of implicitly programming the robot, but it also makes the learning problem become significantly more tractable by separating redundancies from important characteristics of a task (Roweis & Saul, 2000). The learning process for multifingered manipulation is shown in Fig. 1.1.

![Figure 1.1: Learning process from human hand motion to multifingered manipulation](image)

Figure 1.1: Learning process from human hand motion to multifingered manipulation

Human hand motion recognition is becoming a field of great interest and benefit to a number of research areas and applications especially in multifingered robot manipulation. However, due to the dexterity and complexity of the human hand, hand motion recognition is still an open problem though it has been investigated over the past two decades. Most existing methods can only achieve satisfactory results with sufficient training samples, which are difficult, sometimes impossible, in realistic scenarios. The real-time issue has been one of the bottlenecks for research development and practical implementation of human hand motion recognition (Huang & Pavlovic, 1995; Mitra & Acharya, 2007; Moni & Ali, 2009; Wu & Huang, 1999, 2001). Different subjects lead to a higher requirement for a good robustness of the recognition methods due to the different personal issues such as habit and practice. In addition, complex human hand motions such as in-hand manipulation have not been addressed (Kruger *et al.*, 2007).
Based on the background, the goal of this thesis is to develop a set of new methodologies and techniques for human hand motion recognition to deal with the problems and challenges which are summarized in Section 1.2.

1.2 Problems and Challenges

Human motion recognition is a fascinating ability learned day by day from childhood, which is an easy and effortless task for human beings in the daily life. However, it is difficult and challenging to transfer similar abilities to computers. Human hand motion involves both spatial (hand pose in each time instant) and temporal (the transition of the hand poses over time) characters in the representations. To develop a long-standing methodology and a feasible solution, the following problems and challenges need to be addressed:

1. Existing recognition methods are confronted with the limitations of restricted subjects, constrained motions and specific training samples. Different subjects and various motions lead to a fundamental challenge that many algorithms have to face in both theoretical and practical development. In addition, one of the biggest challenges for researchers is how to overcome the limitation of a huge volume of training samples required by the learning algorithms, which is an important criterion for evaluating the training speed.

2. Mature statistical models, e.g. Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and their invariants, have been widely used to model human hand motions. However, the slow training processes limit their applications especially in real-time systems, and the trade-offs or balance between computational cost and recognition accuracy have to be handled. Combining statistical models with fuzzy approaches brings a promising way to improve the learning efficiency, but it is still at the early stage.

3. In-hand manipulation is the ability to change the position/orientation or adjust an object within one hand. It is much more complicated than simple grasp motion and is associated with the most complex human motor skill. There is no
1.3 Overview of Approaches and Contributions

To take into account the previous section, this thesis makes contributions to the problem areas described in Section 1.2. The contributions driven by three novelties are described as follows.

1. This thesis proposes a novel fuzzy framework of a set of recognition algorithms: extended Time Clustering (TC), Fuzzy Gaussian Mixture Models (FGMMs) and Fuzzy Empirical Copula (FEC) that use numerical clustering, Gaussian pattern and data dependence structure respectively for human hand motion recognition. The framework consists of the three recognition algorithms with different level features, which has never been proposed before. In this framework, various hand motions from different subjects can be successfully modeled with limited training samples. It provides an effective solution for human hand motion recognition in different practical applications especially with limited training samples. The extended TC is applicable to motion planning directly transferred from the recognition result. FGMM is applicable to the applications which have a small model storage space and require a method to generate the desired trajectory. FEC can be used in the applications requiring high recognition rate and no desired trajectory with limited training samples.

2. The methodologies in the framework are novel fuzzy approaches, which have been proposed or integrated with statistical models to enhance the learning efficiency in this thesis. An extension of the TC algorithm is proposed by extending the degree of membership with both time instance weights and different learning motion weights. The extended TC not only has the capability to model repeated motions from the same subject but also can learn similar gestures from various subjects. A validation has also been presented, which demonstrates its effectiveness. FGMM is proposed by utilizing a weighting exponent on the fuzzy membership to improve the convergence speed; Two new types of FGMMs based on
the generalized GMMs have been proposed, which are probability based FGMMs and distance based FGMMs. The comparison results of the proposed FGMMs with conventional GMMs and generalized GMMs on various kinds of datasets shows that FGMMs have better efficiency than conventional GMMs and generalized GMMs. It also found that distance based FGMMs outperform probability based FGMMs in terms of learning efficiency. FEC is proposed by integrating high dimensional Fuzzy Clustering by Local Approximation of Memberships (FLAME\(^+\)) with EC to save the computational cost of dependence structure analysis. FLAME is firstly extended into multi-dimensional space, then the FLAME\(^+\) is utilized to reduce the sampling data and maintaining the interrelations at the same time before data dependence structure estimation takes over. FEC is demonstrated to be an effective dependence estimation method with a greatly increased efficiency. These improvements further strengthen the efficiency of the proposed framework in the context of optimally real-time human hand motion recognition.

3. Experimental results have been presented, which compare the methodologies in the framework with HMM and GMM. Various unconstrained human hand motions consisting of both grasp motions and in-hand motions have been employed to test the proposed framework. A wide range of scenarios of human hand recognition are also used to evaluate the proposed algorithms: a) different datasets; b) single subject and multiple subjects; c) varying training samples. It is demonstrated that components of the proposed framework are not only capable of identifying grasp motions but also have the ability to recognize in-hand manipulations, which is the most complex human motor skill and has not been addressed before. In addition, the empirical results have shown that the proposed framework outperforms HMM and GMM in terms of both effectiveness and efficiency criteria for both of grasps and in-hand manipulations.

### 1.4 Outline of Thesis

The rest of the thesis is organized as follows.
Chapter 2 presents a survey of the most recent work on human hand motion analysis with an emphasis on the methods for human hand motion recognition. The aim is to provide readers a systemic and comprehensive understanding of the current developments in human hand capture, human hand models and recognition methods. The concluding discussion assesses the progress thus far and outlines some research challenges and future direction, as well as the solutions to which is essential to achieve the goals of human hand motion analysis.

Chapter 3 introduces TC algorithm, a fuzzy time-modeling approach, based on fuzzy clustering and Takagi-Sugeno modeling. Different from the standard TC algorithm, the degree of membership of the algorithm is extended with both time instance and different learning motion weights. Then, a norm function is employed to recognize human hand motions that is capable of instinctively identifying the start point and end point of the motion with numerical value as output. The improved TC has a better performance of modeling repeated motion from the same subject or similar motions from various subjects.

Chapter 4 first proposes generalized GMMs by integrating conventional GMMs and active curve axis GMMs for fitting non-linear datasets. Two types of FGMMs are then proposed based on the generalized GMMs: probability based FGMMs and distance based FGMMs. It is demonstrated that the proposed FGMMs not only possess non-linearity to fit datasets with curve manifolds but also have a much faster convergence process saving more than half computational cost of GMMs. In the end, FGMMs based recognition algorithm for human hand motion is proposed.

Chapter 5 presents FEC which integrates Fuzzy Clustering by Local Approximation of Memberships (FLAME) and EC to reduce the computation time of dependence structure estimation, which is the structure of dependence relations. In FEC, The highest density objects are first identified to represent the original dataset and then its dependence structure is quickly estimated. Two case studies have been carried out to demonstrate that FEC can substantially reduce the computational cost while features of the data are maintained. The algorithm of hand motion recognition using FEC is finally presented with a motion template and a dissimilarity function.

Chapter 6 demonstrates a thorough investigation on the proposed framework which includes a wide range of scenarios of human hand recognition: 1) different datasets
including 13 grasps and 10 in-hand manipulations. 2) single subject and multiple subjects. 3) varying training samples. The performance of the proposed fuzzy framework is compared with GMM and HMM under the same conditions to demonstrate its effectiveness and efficiency.

Chapter 7 first discusses and summarizes the experiment results, the proposed framework and main contributions of the thesis. Both the strengths and weaknesses of the proposed algorithms are discussed. Finally, future research work is identified, addressing the weaknesses of the approaches and hypothesizing on how to further apply the developed methodologies.
Chapter 2

Literature Review

2.1 Introduction

The human hand and brain are two of the most distinguished features which differ from other animals. According to several findings of paleoanthropologists, some philosophers concluded that the mechanical dexterity of the human hand has been a major factor in allowing homo sapiens to develop a superior brain (that is to say, a similar role played by the anatomical structure of the human larynx in relation with speech capabilities has been also recognized) (Bicchi, 2000).

There are 27 bones within the wrist and hand, shown in Fig. 2.1. The wrist itself contains eight small bones called carpals. The carpals join with the two forearm bones, the radius and ulna, forming the wrist joint. Further into the palm, the carpals connect to the metacarpals. There are five metacarpals forming the palm of the hand. One metacarpal connects to each finger and thumb. Small bone shafts called phalanges line up to form each finger and thumb. The main knuckle joints are formed by the connections of the phalanges to the metacarpals. These joints are called the metacarpophalangeal joints (MCP joints). The MCP joints work like a hinge when you bend and straighten your fingers and thumb. The three phalanges in each finger are separated by two joints, called interphalangeal joints (IP joints). The one closest to the MCP joint (knuckle) is called the proximal IP joint (PIP joint). The joint near the end of the finger is called the distal IP joint (DIP joint). The thumb only has one IP joint between the two thumb phalanges. The IP joints of the digits also work like hinges when you bend and straighten your fingers and thumb. The joints of the hand, fingers,
2.1 Introduction

Figure 2.1: Bones of the human hand and forearm. Individual bone names are underlined. The metacarpal, proximal phalanx, and distal phalanx bones exist in each finger of the human hand, while the middle phalanx bones exist in all fingers but the thumb (Albrecht et al., 2003).

and thumb are covered on the ends with articular cartilage. This white, shiny material has a rubbery consistency. The function of articular cartilage is to absorb shock and provide an extremely smooth surface to facilitate motion. There is articular cartilage essentially everywhere that two bony surfaces move against one another, or articulate.

Every day human hands perform a huge amount of dexterous grasps to fetch, move and use different objects very easily due to the innate sense for goal attainment and sensorimotor control. However, these tasks are relatively difficult for a multifingered robotic hand because of the lack of appropriate sensor systems and some unsolved problems with the human-robot interaction (HRI). Though an artificial hand may perform stronger and faster grasps than the human hand, the high dimensionality makes it hard to program and manipulate human-like robotic hand for dexterous grasps as a human does.

Though great improvements in the hardware of robotic hands have been achieved
2.1 Introduction

during the past decades, such as Robonout hand, Southampton Hand et al. (Bicchi, 2000; Light & Chappell, 2000; Lotti et al., 2005), requirements of the human-like robotic hand, such as advanced sensory systems and practical human-robot interaction technology, are still far away from being satisfied. Recently, due to significant innovations in multifingered robot hands and mature algorithms in robot planning (Latombe, 1991; Long & Fox, 2003), priority has been given to multifingered robot object manipulation. Computer scientists have made significant advances in computational intelligence for robot manipulation (Al-Gallaf, 2006; Fuentes & Nelson, 1998; Schilling & Cruse, 2007; Simmons & Demiris, 2006). Gomez et al. (2005) developed an adaptive learning mechanism, which allows a tendon driven robotic hand to explore its own movement possibilities, to interact with objects of different shapes, sizes and materials and to learn how to grasp and manipulate them. Liu et al. proposed heuristic algorithms to generate desired trajectories for multifingered robots (Liu & Dai, 2002; Liu & Lin, 2008).

However, the manipulation systems of the robotic hands are hardcoded to handle specific objects by their corresponding robot hands. It is evident that robot hand control and optimisation problems are very difficult to resolve in mathematical terms, however humans can solve their hand manipulation tasks easily using skills and experiences. Object manipulation algorithms are needed that have human-like manipulation capabilities and are independent of robot hand hardware. Hence, the main challenge that researchers now face is how to enable robot hands to use what can be learned from human hands, to manipulate objects, with the same degree of skill and delicacy as human hands. For instance, given the locations and shapes of a cup by off-the-shelf image processing algorithms, a robotic hand is required to reach and manipulate the cup by continuously responding with appropriate shape configuration and force distribution among the fingers and palm, inspired by human hand biological capabilities.

Human-robot skill transfer provides a feasible way to solve the hard-coded problem and is important to develop advanced multifingered manipulation planning and control systems. Zoliner et al. (2005) and Chella et al. (2004) suggest prior knowledge be introduced to achieve fast and reusable learning in behavioural features, and integrating it in the overall knowledge base of a system is helpful to fulfill the requirements such as reusability, scalability, explainability and software architecture (Jacobsson et al., 2008; Verschure et al., 2003). It is also showed that characteristics of human grasping could
2.1 Introduction

Enhance dexterity in Robotic Grasping (Bae et al., 2005). In addition, Robertsson et al. (2007) and Carrozza et al. (2006) demonstrated that human-like functionality could be achieved even if the structure of the system is not completely biologically inspired. Unlike teleoperation-related methods, learning from human hand motions is preferred for human-robot skill transfer in that it provides non-contact skill transfer from human motions to robot motions by a paradigm that can endow artifacts with the ability for skill growth and life-long adaptation without detailed programming (Calinon et al., 2007). In principle, not only does it provide a natural, user-friendly means of implicitly programming a robot, but it also makes the learning problem become significantly more tractable by separating redundancies from important characteristics of a task (Roweis & Saul, 2000).

Figure 2.2: The main fundamental phases of the learning process for multifingered manipulation

In this chapter, the learning process for multifingered manipulation is discussed from the three main fundamental phases shown in Fig. 2.2 in Section 2.2, 2.3 and 2.4. Modeling human hand motion capabilities involves understanding the motion of a highly articulated and constrained human hand containing 27 bones giving it roughly 25 degrees of freedom or ways of moving. Dependences among fingers and joints make modeling hand motion even more complex.

Generally speaking, there are four types of methods for both static and dynamic whole hand modeling in terms of sensory data capture: (i) biological hand modeling; (ii) hand motion recognition based on motion capture devices (e.g., data gloves); (iii) vision based modeling and (iv) EMG-based modeling. The performances of these devices depend on capabilities of their sensors.

Computational intelligence methods, such as HMMs, GMMs and other connectionist approaches, have made a significant contribution to the human motion recognition. This Chapter presents a survey of most recent work in the aspects including...
hand motion capture, hand model, recognition methods. This provides a systemic and comprehensive understanding of the most recent work in this field, and identifies the open areas for future research.

The remainder of this chapter is organized as follows. Section 2.2 is devoted to hand gesture capturing based on glove, vision and electromyography (EMG). Section 2.3 presents the methods of hand modeling including hand shape model, kinematic hand model and dynamic hand model. Section 2.4 discusses various recognition methods, with particular emphasis on HMM, Finite State Machine (FSM), and Connectionist approaches. The last Section concludes this Chapter and indicates some existing challenges and future research possibilities.

2.2 Hand Motion Capture

Figure 2.3: Hand motion capture of the learning process for multifingered manipulation

Hand motion capturing shown in figure 2.3, obtaining a hand gesture, is the first step to import human manipulation skills. Sufficient and accurate acquired data is the precondition of good training results. Normally, three capturing equipments are mostly used, which are data glove, camera, and bioelectric sensors.

2.2.1 Glove based Capturing

Data glove is one of the most important input device for analyzing the hand configuration, which measures the movements of the wearer’s fingers and transmits them to the computer. Various sensor technologies, e.g. magnetic tracking devices and inertial tracking devices, are used to capture physical data such as bending of fingers, finger positions, acceleration, force and even haptic feedback. The first data glove, the Sayre
2.2 Hand Motion Capture

Glove, was developed in 1977 by Thomas de Fanti and Daniel Sandin. It used light based sensors, which are flexible tubes with a light source at one end and a photocell at the other. As the fingers with tubes were bent, the amount of light that hit the photocells varied, thus providing a measure of finger flexion. After that, numerical gloves have been proposed using different sensor technologies over the last three decades. The most popular sensors used in data gloves include piezoresistive sensor such as P5 Glove (Sturman et al., 1994) and CyberGlove (Burdea & Coiffet, 2003; Sato et al., 2004), fiber optic sensor such as SpaceGlove (Sturman, 1992) and 5DT Glove, hall-effect sensor such as Humanglove (Dipietro et al., 2003), and magnetic sensors such as 3d Imaging Data Glove (Su et al., 2003) and StrinGlove (Kuroda et al., 2004). They translate the fingers inflection or abduction movements precisely into electric signals at a fast speed. Dipietro et al. (2008) presented a good review on data glove systems and their applications. Here the focus is on the more recent data gloves available in the market, shown in the figure 2.4. These gloves adapt more sensors including bend sensors, touch sensors and accelerometers to measure not only finger angles but also finger abductions, pressure applied and hand tilt. They are predominant in the competitive market due to their high precision at a high speed, and their parameters are listed in table 2.1. In addition, their considerate accuracy and easy control have gained a number of interests.

Cyberglove is chosen in this thesis because of its good function and service support. The Cyberglove has the capability of capturing finger and palm angles using 22 sensors, which is enough for recording human hand motions in the experiments. Its high sensor rate, whose minimum is 90 Hz per second, provides a reliable data source covering both the slow motions and fast motions. More importantly, directly capturing the angle data of the hand prevents the users from additional image processing and object tracking. In addition, the commercial support from the company provides a friendly user interface and mature data storage system, which will improve the experiment efficiency and secure the data storage.

However, most of the commercial data gloves are very expensive, for example X-IST Data Glove HR3 would cost about $4200.00 currently. Nissho electronics’ SuperGlove (LaViola, 1999) has same features as CyberGlove but is much cheaper, although its senses are quite limited in the number of degrees of freedom. Thereafter, more attention has been paid to invent a cheap data glove with high performance. Kuroda
### 2.2 Hand Motion Capture

<table>
<thead>
<tr>
<th>Device</th>
<th>Technology</th>
<th>Sensors and locations</th>
<th>Precision</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG5-VHand</td>
<td>accelerometer and piezo-resistive</td>
<td>6 (a 3 axes accelerometer in wrist; one bend sensor per finger)</td>
<td>10 bit</td>
<td>25 Hz</td>
</tr>
<tr>
<td>5DT Glove 14</td>
<td>fiber optic</td>
<td>14 (2 sensors per finger and abduction sensors between fingers)</td>
<td>8 bit</td>
<td>Minimum 75 Hz</td>
</tr>
<tr>
<td>X-IST Glove</td>
<td>piezo-resistive, pressure sensor and accelerometer</td>
<td>14 (Sensor selection: 4 Bend sensor lengths, 2 pressure sensor sizes, 1 two-axis accelerometer)</td>
<td>10 bit</td>
<td>60 Hz</td>
</tr>
<tr>
<td>CyberGlove II</td>
<td>piezo-resistive</td>
<td>22 (three flexion sensors per finger, four abduction sensors, a palm-arch sensor, and sensors to measure flexion and abduction)</td>
<td>8 bit</td>
<td>minimum 90 Hz</td>
</tr>
<tr>
<td>Humanglove</td>
<td>hall-effect sensors</td>
<td>20/22 (three sensors per finger and the dbduction sensors between fingers)</td>
<td>0.4 deg</td>
<td>50 Hz</td>
</tr>
<tr>
<td>Shapehand glove</td>
<td>bend sensors</td>
<td>40 (flexions and adductions of wrist, fingers and thumb)</td>
<td>n.a.</td>
<td>100 Hz maximum</td>
</tr>
</tbody>
</table>
et al. (2004) presented an innovative intelligent consumer priced data-glove named StrinGlove, which obtains full DoFs of human hand using 24 inductcoders and 9 contact sensors, and encodes hand postures into posture codes on its own DSP, shown in Fig. 2.5. Another data glove, which has the capability of measuring ten DoFs of a hand with only five sensors arranged properly on the palmar surface instead of the dorsal surface, was presented by Fahn & Sun (2005). Moreover, Pamplona et al. (2008) proposed the idea and the prototype of Image-Based Data Glove, which uses a camera to track visual markers at finger tips and is possible to turn into a low cost data glove. Gentner & Classen (2009) developed a low cost but powerful WV-glove at university of Wuzburg, which adapts 14 sensors to measure flexion and abduction movements.

Due to the improvements of the sensor technology, glove base system is more used in robotics (Aleotti & Caselli, 2006; Brown & Asada, 2007; Heumer et al., 2008; Liu et al., 2007; Tegin et al., 2009). Palm et al. (2009) and Ju et al. (2009) use CyberGlove to record the human grasp data to build up different human grasp models. A data glove equipped force feedback is adapted for training multi-fingered robot in
2.2 Hand Motion Capture

(Kawasaki et al., 2009). Not only could the fingers’ angles be recorded in the human hand skill models, but also the finger tips haptic information and the positions can be integrated in the models through data gloves (Alhalabi et al., 2004; Ceruti et al., 2009; Shen et al., 2003). Wang et al. (2008b) designed a haptic data glove (TSG gloves) based on the Tekscan grip sensor 4255N to model human handshake skills. The TSG glove has good performance in measuring interaction forces especially during hand-shaking. Romano et al. (2009) introduced the SlipGlove which provides tactile cues associated with slip between the glove and a contact surface. These gloves with haptic-IO capability provide the vital information of human hand motion and greatly enhance the capturing of human hand skills.

Generally speaking, before using the data gloves, a time-consuming calibration phase is needed in order to account for differences in hand size and proportion when mapping from raw sensor readings to joint angles of the user’s hand. How an optimal calibration can be achieved is still an unsettled question. Due to cross-couplings between the sensors, more complex forms of calibration are necessary to achieve a satisfactory fidelity. Kahlesz et al. (2004) developed a calibration method that explicitly models the cross-couplings of the abduction sensors with the neighboring flex sensors without any auxiliary calibration hardware. Some intelligent methods are also employed to calibrate data gloves, for example, Sun et al. (2006) presented a calibration method based on genetic algorithm which is able to avoid getting trapped in local
2.2 Hand Motion Capture

extremum and setting initial values.

2.2.2 Vision based Capturing

Data gloves developed so fast that they became wireless and more comfortable, yet they are not as convenient and practical as cameras as they need connection to computers. Since the gloves always cling to hands, haptic sensing of the hand become ineffective and volume of finger activity is usually limited. The glove is connected by the wire or heavy transmitter, which would cause extra problems with the manipulations. Moreover, it is time consuming to calibrate a data glove and get an experiment setup for a user.

Using computer vision to track human movement and extract data of gestures frees the movement of the demonstrator in a much more natural and non-contact way. It is not necessary for cameras to be calibrated and set up every time for different users. However, to collect data from a vision sensor, the hand has to be localized in the image sequences and be segmented from the background. Because a hand is highly articulated and constrained, both single cameras and multiple cameras have been used to achieve both 2D and 3D hand gesture models.

2.2.2.1 Single Camera

One camera is used for 2D hand models, which can also be called appearance-based models (Ge et al., 2008). The position of the camera should guarantee good visibility and it is better to avoid some areas in which some of the fingers are shadowed. As a sequence of views, the gestures are modeled by associating the images of hand gestures to the appearance of predefined ones. Since 2D hand models are restricted to a camera’s viewpoint, significant efforts had been paid to depict the geometric structure of hand 3D models using one camera. This approach searches for the kinematic parameters which map the 2D projection images to the 3D hand models (Athitsos & Sclaroff, 2003; de La Gorce et al., 2008; Lee & Cohen, 2004; Segen & Kumar, 1999; Stenger et al., 2007).

Due to the fact that it is not a trivial task to separate the hand from complex backgrounds, range cameras are used intensively to achieve depth images in which a hand can be segmented easily. Range cameras can operate according to a number of different
2.2 Hand Motion Capture

techniques including stereo triangulation, sheet of light triangulation, structured light, time-of-flight, interferometry, coded aperture and so on (Bray et al., 2004b; Breuer et al., 2007; Liu & Fujimura, 2004; Malassiotis & Strintzis, 2008). For instance, Bray et al. (2004b) applied range data using structured light for hand gesture recognition, while Liu & Fujimura (2004) and Breuer et al. (2007) used infrared range data from a time-of-flight camera for real-time hand gesture recognition.

2.2.2.2 Multiple Cameras

Though using only one camera can reduce the complex computation, it is sensitive to viewpoint changes and constrained environments such as plain backgrounds or controlled lighting, and it cannot provide precise spatial information. Using more than one camera to capture human hand gestures can overcome the difficulties such as self-occlusion and wider range of poses and get more reliable 3D hand configuration (Abe et al., 2000; Bendels et al., 2004; Jennings, 1999; Sato et al., 2001; Utsumi & Ohya, 1999). Combining multiple view information can be roughly classified into three categories: low-level combination, high-level combination and intermediate-level combination. The low-level approach is to group all the shape contexts from all the images together before clustering to build the histograms, for instance, Ueda et al. (2003) reconstructed the hand pose as a “voxel model” by integrating multi-viewpoint silhouette images from a multi-viewpoint camera system. The high-level combination is to estimate the pose from each view individually and combine the results at a high level using, for example, a graphical model. The approach proposed by de Campos & Murray (2006) combined the information at an intermediate level. Description vectors are generated for each camera individually and then concatenated into a higher dimensional vector describing the current measurements from all the cameras.

Using only cameras may lead to complex combination computation, so markers are introduced to help the cameras capture the configuration much faster and more precisely. Kim & Fellner (2004) presented a hand gesture interaction system for a back projection wall environment, using thimble-shaped fingertip markers made of white printing paper with a ‘black light’ source. Gesture parameters are calculated by 3D positions of the marked fingertips and their pinching states. Usabiaga et al. (2005) used more than 2 cameras to track the 3D position and orientation of an elliptical
2.2 Hand Motion Capture

marker placed on the dorsal part of the hand using model-based tracking approaches and active camera selection.

2.2.3 Electromyogram based Capturing

The camera can get sufficient information of human hand joint angles or positions, but it is incapable of capturing the force of human hand manipulation. One intuitive way to get force information is to employ the force sensors such as FlexiForce sensors and Finger TPS, which are often attached to data glove. An alternative is to use the Electromyography (EMG) signal from the muscle to estimate the applied force. With the developments in the biosignal exploration, EMG has gained more and more interests especially in studying the human hand manipulation.

Electromyography (EMG) is a technique for evaluating and recording the activation signal of muscles, and it includes surface EMG (sEMG) and needle EMG (nEMG)/intramuscular EMG (iEMG). sEMG is a non-invasive approach to measure muscle activity where surface electrodes are placed on the skin overlying a muscle or group of muscles. Because of the advantages of non-invasion and easy operation, sEMG has been used intensively in the hand movement studies. Fukuda et al. (2003) used sEMG signals to determine the joint of the controlled human-assisting manipulator. Reddy & Gupta (2007) proved that the sEMG signals from the flexor digitorum superficialis and
2.3 Hand Modeling

Composed of 27 pieces of bones including 14 phalanges, 5 metacarpals and 8 carpals, the human hand has complex kinematics and highly articulated mechanism with more than 20 degree of freedoms. Natural anatomical restrictions subject to the muscle-tendon controlling mechanism make the hand modelling more precise and at the same time even harder (Jones & Lederman, 2006). This section as shown in figure 2.7 gives
2.3 Hand Modeling

Figure 2.7: Hand modeling of the learning process for multifingered manipulation

a review of human hand modelling in terms of kinematic structure model, static model and dynamic model.

2.3.1 Kinematic Hand Model

Based on the hand anatomy, a kinematical model with 27 Degrees of Freedom (DOF) was introduced in (Lee & Kunii, 1995; Lin et al., 2000). Each of the four fingers has four DoFs. The distal interphalangeal (DIP) joint and proximal interphalangeal (PIP) joint each has one DoF and the metacarpo-phalangeal (MCP) joint has two DoFs due to flexion and abduction. The thumb has a different structure from the other four fingers and has five DoFs, one for the interphalangeal (IP) joint, and two for each of the thumb MCP joint and trapeziometacarpal (TM) joint both due to flexion and abduction. The hand totally has 21DoFs. The remaining 6 DoFs are from the rotational and translational motion of the palm with 3 DoFs each. Some other kinematic models with different numbers of DOFs can be seen in (Kuch & Huang, 1994) which gave the palm two internal DOF located at the base of the fourth and fifth (ring and pinky) metacarpals, (Dragulescu & Ungureanu, 2007) which ignored the translation motion of the palm, (Bray et al., 2004a,b) which introduced an extra twist motion to MCP joints, and (Nirei et al., 1996) which added one flexion/extension DOF to CMC joints.

The kinematic hand models discussed above are well matched to the true geometry of the hand, but there are a few simple models for the fast and effective computation in some particular environments. Sudderth et al. (2004) introduced a kinematic model in which nodes correspond to rigid bodies and edges to joints. Heap & Hogg (1996) constructed a 3D deformable Point Distribution Model of human hand surface instead of a kinematic model.
2.3 Hand Modeling

The models discussed cannot represent the whole hand kinematical structure without constrains. One of the advantages of considering the motion constraints among fingers and finger joints is to greatly reduce the size or dimensions of the search space. Another one is to simulate natural hand motion and produce more realistic hand animation, which would be of a great help to robotic hand manipulation and synthesis of the sign languages. Lin et al. (2000) introduced three types of constraints, which are limits of finger motions (see equations 2.1 2.2 and 2.3) referred as static constraints, limits (see equations 2.4) referred as dynamic constraints imposed on joints during motion, and limits in performing natural motion. Some of these common constraints also can be seen in (Bray et al., 2004b; Kuch & Huang, 1994; Pavlovic et al., 1997).

\[
0^\circ \leq \theta_{MCP-F} \leq 90^\circ, \\
0^\circ \leq \theta_{PIP} \leq 110^\circ, \\
0^\circ \leq \theta_{DIP} \leq 90^\circ, \text{and} \\
-15^\circ \leq \theta_{MCP-AA} \leq 15^\circ \\
\theta_{MCP-AA} = 0 \\
\theta_{TM-AA} = 0 \\
\theta_{DIP} = \frac{2}{3}\theta_{DIP}
\]

(2.1) (2.2) (2.3) (2.4)

where the subscript $F$ denotes flexion and $AA$ denotes abduction or adduction.

Chua et al. (2002) reduced 27 DoFs of the human hand model to 12 DoFs by analyzing the hand constraints which were divided into eight different types. In addition, Chua et al. analyzed the constraints proposed by Rijpkema & Girard (1991), Kuch & Huang (1995), and Lee & Kunii (1993). Even more, he introduced four other constraints such as “finger plane”, “thumb plane”, and “middlefinger plane”. Some of these eight constraints may be “weak constraints” in static analysis or dynamic analysis. However, the failure of these constraints may lead to an invalid solution.

Due to the advantages of considering motion constraints, some of them are often employed in current research such as animation and visual tracking (Lin et al., 2000; Schieber & Santello, 2004). The computational complexity of hand analysis may be reduced significantly with the consideration of such constraints, but a large number of
hand motion constraints are very hard to describe in closed forms and may cost extra computational cost. Therefore how to add suitable constraints to the hand analysis should be considered carefully according to the specific purpose.

However, it’s difficult to explicitly represent the constraints of natural hand motions in closed form, some of which can not or have not been found. Lin et al. (2000) introduced a model whose constraints were learnt from a large and representative set of training samples.

2.3.2 Static Hand Model

2D hand models mostly depend on the extracted image parameters, which are derived from the hand image properties including contours and edges, image moments, image eigenvectors and other properties. Du & Li (2000) achieved stable detection by extracting two kinds of features, which are statistic-based feature and contour-based feature with only one camera, although the hand was moved on a plane, which produced good visibility. Eigenvalues indicating the hand width and length were extracted by Binh et al. (2005), who used one camera to build a hand gesture recognition system for real time America Sign Language in unconstrained environments, while Haar-like features are extracted from 2D hand images to recognize hand gestures in (Chen et al., 2007). Wu et al. (2001) used a cardboard model in Fig. 2.8, which is a simplification of the real hand and offers a good approximation for motion capturing under this specific

![Figure 2.8: Cardboard hand model (Wu et al., 2001)](image)
view direction. When viewed from the direction orthogonal to the palm, the hand could be modeled by a cardboard model, in which each finger could be represented by a set of three connected planar patches. Other features such as ‘bunch graph’ in (Tríescher & von der Malsburg, 2002), ‘conducting feature points’ in (Je et al., 2007) and ‘grid of neurons’ in (Stergiopoulou & Papamarkos, 2009) were also extracted from 2D images. 2D hand models are robust to self-occlusion since they extract no 3D features and directly compare the 2D image properties between the input images and the registered ones, but images without shadowed fingers can enhance the validity of 2D hand models.

3D hand models may lead to the computational complexity caused by inverse kinematics and 3D reconstruction. How to get 3D geometric configuration with only one camera and avoid the complex computation was studied. In (Shimada et al., 2001), an appearance database was built up which contained more than 16,000 possible silhouette contours generated from a given 3D shape model by rotating model joints. For real-time processing, the search area was reduced by using an adjacency map in the database. By retrieving the appearance in the database which matched the input image contour, the joint angles of the input shape can be rapidly obtained. Stenger et al. (2001) built an anatomically accurate hand model from truncated quadrics as shown in Fig. 2.9. This allows for the generation of 2D profiles of the model using elegant tools from projective geometry, and for an efficient method to handle self-occlusion. In addition, Chua et al. (2002) analyzed some of the hand’s constraints to reduce the 27 to 12 DOFs without any significant degradation of performance, and proposed a
novel algorithm to estimate the 3D hand posture from the eight 2D projected feature points at a very high speed. Color markers are employed to identify these eight points. In addition to these rough hand models, which make processes efficient, some more complex models are built for better appearance. In order to model the hand surface, Dewaele et al. (2003) used one spheroid-based per part (palm or phalanx), and a total of 16 meta-balls for the hand. Four different methods were built: using spheroids only, fusing all the spheroids using soft objects, fusing spheroids belonging to the same finger, and fusing each spheroid with its neighbors in the structure, as shown in Fig. 2.10. Bray et al. (2004b) employed a deformable model which consists of a polygonal skin, driven by an underlying skeleton, and reproduces actual hand shapes quite well, shown in Fig. 2.11. On the other hand, since the single camera is sensitive to the environmental conditions e.g. background, illumination etc. and has the problems of distortion and ambiguities, 3D hand model has been intensively studied using multiple cameras (Bendels et al., 2004; Kim & Fellner, 2004; Usabiaga et al., 2005).

### 2.3.3 Dynamic Hand Model

The above models can be seen as static models since there is no temporal information in these models. Dynamic hand model uses both the temporal and shape characteristics as the features are extracted. The dynamic aspect is defined either by the trajectory of the hand, or by a series of hand postures in a sequence of images (Just & Marcel, 2009). Different with static hand model, building dynamic hand model needs to solve the problem of spotting. Spotting aims to identify the beginning and/or the end of a
Figure 2.11: The hand model as polygonal surface (top). The hand model and its degrees of freedom (bottom) (Bray et al., 2004b).

motion given a continuous stream of data which is a random sequence containing both known and unknown motions. It is difficult because:

- Concerning temporal difficulty, repeats of same motions may take different times. It’s almost impossible to exactly repeat the same motion twice.

- Concerning spatial difficulty, hand grasps switch from one motion to another when moving. Some intermediate postures may be mistaken as start or end points due to the same configurations.

These difficulties are illuminated in (Mitra & Acharya, 2007) as ‘the segmentation ambiguity’ and ‘the spatio-temporal variability’. Some related research shows that these difficulties can be solved efficiently by fusion of additional information sources such as tactile sensing and vision (Bernardin et al., 2005; Ekvall & Kragic, 2005). Palm et al.
(2009) proposed a more general segmentation method which performs a convolution of a test sequence of grasps against a model signal, plus a subsequent time clustering. After segmentation, how to determine class membership, which map the sampling motions to the models, and how to implement the recognition procedure should be considered. These memberships depend on the choices of recognition methods. Different recognition methods may adopt different memberships, which are important to evaluate the specialties of these recognition methods. The methods for dynamic models include Hidden Markov Models (HMM), Finite state Machine (FSM) and so on, which will be discussed in Section 2.4.

2.4 Hand Motion Recognition

Figure 2.12: Hand modeling of the learning process for multifingered manipulation

Human hand motion recognition as shown in figure 2.12 is attracting more and more research interests as it benefits for a number of research areas and applications especially in multifingered robot manipulation. Human hand motions are naturally recognized firstly by segmenting the gesture sequences. For instance, the start and end points of a grasp motion should be specified in terms of time and space. And then human compares the specified partitions with ‘experience’ in his mind, and determines the grasp types. To make computers recognize the grasps, motion spotting indicates specifying the start and end points of a meaningful motion from a continuous stream of input signals, and results in segmentation of the relevant motions. However, the spatial-temporal variability also lead to the difficulties both in space and time.

After this, there are various tools for human hand motion recognition, based on approaches ranging from statistical modeling, computer vision and pattern recognition, image processing, connectionist systems, etc. Most of the problems have been addressed based on statistical modelling, such as HMMs in Sec. 2.4.1, and GMMs in
Sec. 2.4.4, and Finite State Machine (FSM) in Sec. 2.4.2 has also been effectively employed in modeling human motions (Bernardin et al., 2005; Hong et al., 2000).

2.4.1 Hidden Markov Models

Markov process is a simple stochastic process in which the distribution of current states depends only on some most recent states and not on all of the past states. The first order Markov process is defined, when the current event depends solely on the most recent past event. When a finite number of states have the property of Markov process, these states can be termed Markov chain. It can also be said that a Markov process is a process or simulation that satisfies a Markov property.

A HMM (Rabiner, 1989; Yamagishi, 2006) is a Markov chain which generates a sequence of discrete time observations. At each time unit, the HMM changes states in accordance with a state transition probability, and then generates observational data in accordance with an output probability distribution of the current state. Transition probability provides the probability for undergoing the transition between hidden states, and output probability defines the conditional probability of emitting an output symbol from a finite alphabet when given a state.

HMM can be recognized as an ergodic model, where any state can be reached from any other state. Especially, the states can transit to themselves, which means any state in HMM may repeat in some cases. For grasp gesture recognition, the state index transits only from left to right. When the hand has specific hand poses before and after grasps, the first and end states are clear. So the initial distribution of the states is determined. In Fig. 2.13, the first state $s_1$ and end state $s_5$ indicate the start and end poses of the grasps. For every grasp HMM, they are the same. In figure 2.13, $\sum_{j=1}^{N} a_{ij} = 1$

Each grasp is modeled by one HMM, so a sequence of grasps is modeled by a sequence of HMMs. Once a system is described as a HMM, three problems can be solved. The first two are pattern recognition problems: Finding the probability of an observed sequence given a HMM (evaluation); and finding the sequence of hidden states that most probably generated an observed sequence (decoding). The third problem is generating a HMM given a sequence of observations (learning).

HMMs are a class of statistical models useful for analyzing a discrete time series of observations. A grasp motion can be considered as a sequence of epochs, each
2.4 Hand Motion Recognition

Figure 2.13: A five states and four observations left-to-right HMM for grasp recognition

of which is characterized by a motion of distinct hand shapes. HMMs can mostly be applied to grasp gesture recognition in the following way:

1. Initialization: Determine the number of the states, the number of the observations and the initial probability distribution for the hidden states.

2. Training of HMMs: With all the grasp sequences, HMMs can be generated by the iterative expectation-modification procedure (EM) known as the Baum-Welsch method. The training process evaluates a log-likelihood of the trained model during iteration and stops as the change of log-likelihood undergoes a certain threshold.

3. Recognition of the test grasps: Compute the log-likelihood between all HMMs and all the test grasps respectively. Use the Maximum Likelihood classification method to classify the test grasps.

Malassiotis et al. (2002) and Hu et al. (2003) presented static hand gestures and the relative image processing and recognition methods were introduced. In (Zhou et al., 2004), based on Local Orientation Histogram Feature Distribution Model, static Hand Gesture Recognition was achieved. The common drawback of these techniques is that only static posture analysis is performed which means that an ideal time point for analysis of the hand configuration has to be extracted from the demonstration by other means before classification can be done.
However, HMM is able to classify the dynamic sequences. The moving hand is tracked and analyzed by Fourier descriptor (FD) from image sequences, and then the feature vector consisting the spatial and temporal features is processed by HMMs for recognition of the hand gesture in (Chen et al., 2003). Ramamoorthy et al. (2003) proposed a recognition strategy using a combination of static shape recognition, Kalman filter based hand tracking and a HMM based temporal characterization scheme. Similarly, Bernardin et al. (2005) presented a method using HMMs to recognize continuously executed sequences of grasping gestures. Both finger-joint angles and information about contact surfaces, captured by a data glove and tactile sensors, are used. By using the Kamakura taxonomy (Kamakura, 1989), most of all grasps used by humans in everyday life can be classified, increasing the system’s application domain to general manipulation tasks. For 12 grasp classes, and with very little training data, a recognition accuracy of up to 89% for a single-user system and 95% for a multiple-user system could be reached. Just & Marcel (2005) presented a two-handed gesture database to manipulate virtual objects on the screen (mostly rotations) and some recognition experiment using HMMs. Binh & Ejima (2006) introduced pseudo three-dimensional Hidden Markov Model (P3DHMM) to recognize hand motion, which gained higher rate of recognition than P2DHMMs (Binh et al., 2005).

A significant limitation of the HMM is that it cannot handle three or more independent processes efficiently (Oliver et al., 1999). To alleviate this problem, researchers have developed dynamic Bayesian networks (DBNs) as generalization of HMMs (Murphy, 2002). DBNs are directed graphical models of a stochastic process, and can generalize HMMs by representing the hidden and observed states in terms of state variables, which can have complex interdependencies. The interdependencies among the state variables can be efficiently represented by the structure of the directed graphical models (Aggarwal & Park, 2004).

HMM is a type of statistical model widely used in many fields, so it has the advantages of incorporation of prior knowledge, the ability of being combined into larger HMMs and mathematical analysis of the results and processes. On the other hand, HMMs are computationally expensive and require large amount of training data. However more training is not always good because it may lead to the over-fitting problem (Cawley & Talbot, 2007). Moreover, HMM may converge to local maxima instead of the truly optimal parameter set for a given training set.
2.4.2 Finite State Machine

A FSM or finite state automaton (plural: automata) or simply a state machine is a model of behavior composed of a finite number of states, transitions between those states, and actions that occur in each state (Lee & Yannakakis, 1996). The number of states may vary for different applications. A FSM can be represented by a state transition diagram, a directed graph whose vertices correspond to the states of the machine and whose edges correspond to the state transitions; each edge is labeled with the input and output associated with the transition. The continuous stream of data for a gesture, from the sensors such as cameras and gloves, are represented as a sequence of states. By this way, a number of samples of each gesture are used as the training data to decide the parameters of every state in the FSM. Then the recognition of gestures can be performed with the trained FSM. According to the input data, the trained FSM recognizer decides whether to stay at the current state or move to the next state. As long as the final state is achieved, the recognition for this gesture has been done. In some cases, for the same gesture, there is more than one model which reaches the final state at the same time. To decide the probable gesture, the membership in a state, which shows how well the state models match the gestures, some choosing criteria can be applied.

Hong et al. (2000) developed a technique for gesture modeling and recognition using FSM in real-time, interactive environments. 2D head and hand locations are tracked repeatedly to be the training data. Considering spatial and temporal information separately, the algorithm first learns the distribution of the data without temporal information via dynamic k-means (Hartigan & Wong, 1979b) and then learns the temporal information from the aligned data segments by the spatial information learning. After the spatial information is then updated, the final state sequence is produced, which represents the gesture. Each state sequence is a FSM recognizer for a gesture. The technique has been successfully tested on a set of gestures, e.g., waving left hand, waving right hand, drawing a circle, drawing a figure of eight, etc.

Yeasin & Chaudhuri (2000) employed a FSM modeling for recognition of the dynamic hand gesture, which helped in interpreting the gesture accurately and also avoided the computationally intensive task of image sequence warping. A novel vision-based system for automatic interpretation of a limited set of dynamic hand gestures
2.4 Hand Motion Recognition

was introduced. The concept of motion energy is used to estimate the dominant motion from an image sequence. They introduced the concept of modeling the dynamic hand gesture using a FSM. The temporal signature is subsequently analyzed by the FSM to automatically interpret the performed gesture.

Stern et al. (2006) tracked and recognized hand gestures for interacting with a video game. Several hand features were extracted and fed into a finite state classifier which identified the hand configuration. The hand gestures could be classified into one of the four gesture classes or one of the four different movement directions.

FSM is the simplest and most basic pattern-recognizer and pattern-describer. It can be used in the real time systems. But the most notable disadvantage is that there is always only one state for the system at any moment. Additionally, the computation spending would rise much if the number of states increases.

2.4.3 Connectionist Approach

Soft computing is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations (Pal & Mitra, 1999). It refers to a collection of computational techniques in computer science, artificial intelligence, machine learning and some engineering disciplines, which attempt to study, model, and analyze very complex phenomena: those for which more conventional methods have not yielded low cost, analytic, and complete solutions. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions (Mitra et al., 2002).

For example, the high dimension raw data from the sensors could be associated with some inherent uncertainties such as the noise and calibration errors, such as the data from a data glove (CyberGlove) has more than 18/22 dimensions and 90 records in every minute. Due to the technology of sensor, the sensor resolution is 0.5 degrees, which means the error is about 0.5 degrees for one angle, and the sensor has about 0.6% nonlinearity over full joint range. In some cases, these uncertainties are unpredictable. To analyze and recognize this data requires much more robust methods which would discover the useful knowledge from the data. Soft computing methodologies for data
mining may be the most promising methods for the recognition problem. It involves fuzzy sets, neural networks, genetic algorithms, and rough sets.

Sato et al. (2001) identified predetermined gestures in a fast and robust manner by using a neural network, which had been properly trained beforehand. A three-layer neural network was employed for the $12 \times 12$ pixels image. Harding & Ellis (2004) applied Fourier analysis to the hand gesture data and then input the complex harmonic data to a Probabilistic Neural Network for gesture classification. In other works, the neural network was trained with a set of training data by using the back-propagation algorithm (Fu, 1994) so that an output node with the highest score represents a recognized hand shape pattern. Several applications were discussed, which includes 3D object handling for a desktop system and 3D walk-through for a large immersive display system. Based on an innovative Self-Growing and Self-Organized Neural Gas (SGONG) network, hand gesture recognition was achieved with recognition rate of 90.5% by Stergiopoulou & Papamarkos (2006).

Chen & Tseng (2007) implemented a system where three support vector machine classifiers were trained for the construction of the hand gesture recognition system. After fusion strategies among these classifiers, the recognition rate increases. Another example is SmartCanvas, an intelligent desk system that allows a user to perform free-hand drawing on a desk or similar surface with gestures, which works based on Support Vector Machine for hand gesture recognition (Mo et al., 2005).

A Modified Census Transform has been successfully used for face detection (Froba & Ernst, 2004). Just et al. (2006) extended this method to hand gesture recognition. A simple linear classifier is trained, using a set of feature lookup-tables. Adding images against complex background in the training set leads to a significant improvement of the recognition rate in complex background conditions.

The fuzzy rule-based method for the recognition of hand gestures is presented in (Bedregal et al., 2006; Natal et al., 2007). The method is highly dependent on a detailed previous analysis of the features of the gestures to be recognized, and on the manual transfer of the results of that analysis to the recognition system. However, Palm et al. used a data glove to capture grasp primitives and modeled fingertip trajectories of grasp primitives by TS-fuzzy models. Fuzzy clustering and modeling of time and space data was applied to the modeling (Palm & Iliev, 2006, 2007). Using sig-
nal processing based on fuzzy nominal scales, dynamic gesture recognition was also presented in (Allevard et al., 2005).

Witten & Frank (2005) presented an approach where a feature vector of hand motion is classified into one of the 10 elementary gestures by a sparse Bayesian classifier. A training set of 628 samples and a testing set of over 1000 samples have been obtained to evaluate the proposed method.

A comparison of classification methods for grasp recognition was studied by Heumer et al. (2007). It introduced a systematic approach for the evaluation of classification techniques for recognizing grasps performed with a data glove. In particular, it distinguishes between 6 settings that make different assumptions about the user groups and objects to be grasped. A large number of classifiers, which are available in Weka data mining software package (Witten & Frank, 2005), was compared to draw informed conclusions about achievable recognitions rates in the different settings. These classifiers are broadly divided into five categories: probabilistic methods (Cooper & Herskovits, 1992; Friedman et al., 1997), function approximators (Bishop, 1995; Haykin, 1994), lazy learners (Bontempi, 2002), trees (Quinlan, 1993), and rule sets (Quinlan et al., 1987; Richards, 2002). From the results of 6 different settings respectively applied on 28 classifiers, the best classifier for every setting is determined.

Each method in soft computing has its own advantage compared with others, which is illustrated in (Mitra et al., 2002). Generally fuzzy sets are suitable for handling the issues related to understandability of patterns, incomplete/noisy data, mixed media information and human interaction, and can provide approximate solutions faster. Neural networks are nonparametric, robust, and exhibit good learning and generalization capabilities in data-rich environments. Genetic algorithms provide efficient search algorithms to select a model, from mixed media data, based on some preference criterion/objective function. Rough sets are suitable for handling different types of uncertainty in data.

### 2.4.4 Other Recognition Methods

Except for the above methods, there are other state-of-art approaches proposed to recognize human hand motions. Miners et al. (2005) presented a graph-based approach to understand the meaning of hand gestures by associating dynamic hand gestures with
known concepts and relevant knowledge. Hand gestures are understood by the comparison between concepts represented by sets of hand-gesture elements and existing knowledge in the form of conceptual graphs using graph morphisms or projection techniques. The conceptual-level processing was emphasized to robustly handle noise and ambiguity introduced during generation, but at the same time it is time-costing. Calinon & Billard (2007) proposed an approach to teach incrementally human gestures to a humanoid robot, which consists of projecting the movement data in a latent space and encoding the resulting signals in a GMM. Kalgaonkar & Raj (2009) used GMMs to classify one-handed gestures based on ultrasonic sensors, and the experiment showed simplified one-handed gestures can be recognized with a high accuracy. However, none of them has addressed the convergence speed of GMMs and the real-time problem is still unsolved.

Fang et al. (2007) took advantage of color and motion cues acquired during tracking to implement adaptive hand segmentation. On the basis of segmentation, multiscale feature extraction is executed and gestures are recognized with palm-finger decomposition. The gestures identified are limited to the finger movements, and only six gestures have been recognized. Five stages of fast and simple recognition method was introduced by Malima et al. (2006) for a limited number of gestures. After localizing, segmentation, finding the centroid and farthest distance and constructing a circle, the final stage Estimates the number of active fingers to the recognition of the gesture.

Chen et al. (2007) proposed a two level approach to solve the problem of real-time vision-based hand gesture classification. The lower level of the approach implements the posture recognition with Haar-like features and the AdaBoost learning algorithm, while the higher level implements the hand gesture recognition using a context-free grammar-based syntactic analysis. Given an input gesture, based on the extracted postures, the composite gestures can be parsed and recognized with a set of primitives and production rules. However, only four gestures were considered in their implementation. In general, these algorithms are only proved to be effective in specific or constrained conditions such as a limited number of gestures and particular subjects. In addition, real-time issue is not addressed enough in some of the algorithms.
2.5 Summary

Learning from human hand motions provides a feasible way to transfer human hand motion skills to robot motions by a paradigm that can endow artifacts with the ability for skill growth and life-long adaptation without detailed programming. Not only does it provide a user-friendly means of implicitly programming the robot, but it also makes the learning problem become significantly more tractable by separating redundancies from important characteristics of a task. In this Chapter, the developments of the three main fundamental phases of the learning process for multifingered manipulation are reported, which are human motion capture, hand modeling and hand motion recognition.

Hand motion capture is fundamental to human hand skill learning. The relationship between an EMG signal and the corresponding motion is not a one-to-one relationship because it varies with time (Kato et al., 2006). So far EMG has just been used for very limited gestures recognition. The accuracy of recognition method based on EMG is lower than two other methods. In many applications, a DataGlove is preferred since it can get data directly, without processing images, which is a hard and complex job for cameras. If the problem of huge computational cost of image processing for feature extracting is solved, the non-contact capture using cameras could have the most potential way to model the human hand skills.

Though there are several alternatives for biological hand modeling, not only is it tremendously difficult to fully model the human hand due to its complex kinematic structure, but also such a detailed model is unnecessary for multifingered robotics. On the other hand, from the modeling perspective, the motion constraints among fingers and finger joints reduce the size and dimensions of the search space, making the estimation of hand postures more cost-effective. The static hand model consisting of 2D and 3D models emphasizes the hand appearance and hand deformation, which provide a visual and straight impression of hand. Dynamic hand model is preferred to store the human hand motion skills since it involves both spatial and temporal information.

Human hand motion recognition is becoming a field of great interest and playing a key role in a number of research areas and applications especially in multifingered robot manipulation. However, due to the dexterity and complexity of the human hand, the recognition of human hand motion involving spatio-temporal variability is still an
open problem though it has been investigated over the past two decades. Most of the existing methods can only achieve satisfactory results with sufficient training samples, which are difficult, sometimes impossible, to get in realistic scenarios, for example the trade-off between computational cost and recognition accuracy have to be handled for HMMs. The slow training processes of the statistic methods, e.g. GMMs, FSM and GMMs have limited their applications especially in real-time systems. In general, the real-time issue has been one of the bottlenecks for research development and practical implementation of human hand motion recognition. Most of the current methodologies do not satisfy the requirements imposed by the different subjects with different personal issues such as habit, practice and anthropometric measurement. In addition, complex human hand motions such as in-hand manipulation has not been addressed. Thus several approaches with different extracted features to solve this problem are considered. TC (Chapter 3), based on the numerical value, is used to model the trajectories using TS fuzzy modeling; FGMM (Chapter 4) uses the Gaussian pattern as the extracted motion feature; FEC (Chapter 5) studies the dependence structure among finger angle values and provides discriminating motion templates to differentiate the motions.
Chapter 3

Time Clustering

3.1 Introduction

Human hand motion recognition is the key for many research areas and applications especially in multifingered robot manipulation and is attracting more and more research interest. However, due to the dexterity and complexity of the human hand, it is still a challenge though it had been investigated in the past two decades. A large number of approaches have been proposed and studied for human hand gesture recognition, but current recognition methods are still facing many problems such as constrained conditions and real-time issues. The speed and precision requirements of the recognition algorithm are difficult to fulfill simultaneously, thus the trade-offs between the speed and accuracy needs to be handled.

To address this problem, Palm et al. (2009) proposed a fuzzy clustering method, time clustering (TC), to recognize human grasps. To model time dependent trajectories using fuzzy modeling, the time instance takes the place of the input variable and the corresponding trajectory points become the outputs of the model. It has been demonstrated that this fuzzy method can speedily represent the dynamic hand grasps by a small number of local linear models and a few parameters (Palm et al., 2009). It can also be used for nonlinear filtering of noisy trajectories and as a simple interpolation between data samples. In addition, it is capable of identifying the start and end points of segments and thus the occurrence of grasps, as well as recognizing the grasp types themselves.
This chapter proposes an expansion to the standard TC algorithm. The degree of membership of the model has been extended with both time instance weights and different learning motion weights. The extended degree of membership has enabled the TC method to be more comprehensive to learn repeated motions from the same subject or similar gestures from various subjects. This chapter is organized as follows: Section 3.2 revisits the fundamental of Takagi-Sugino fuzzy modeling; Section 3.3 introduces the recognition algorithm using Time Clustering, which includes model construction and recognition; A case study is presented to demonstrate its performance comparing with GMM and HMM in Section 3.4. Finally, this chapter is concluded with discussions in Section 3.5.

3.2 Takagi-Sugino Fuzzy Modeling

A fuzzy model has been proposed by Takagi & Sugeno (1985) with fuzzy IF-THEN rules representing local linear input-output relations of non-linear systems. The local dynamics of a system are described with a linear model, and the overall fuzzy model is a combination of this linear model. The $i$th rule of the fuzzy model has the following form:

$$R(i): \text{IF } z_1 \text{ is } M_i^1 \text{ and } z_2 \text{ is } M_i^2, \ldots, \text{ and } z_g \text{ is } M_i^g \text{ THEN } y(t) = A_i x(t) + B_i u(t).$$

where $M$ is the typical fuzzy set and $i$ is the number of rules in the fuzzy model; $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector, $y(t) \in \mathbb{R}^q$ is the output vector, and $z_1, z_2, \ldots, z_g$ are measurable variables. $A_i \in \mathbb{R}^{n \times n}$ and $B_i \in \mathbb{R}^{n \times m}$ are suitable matrices. Assuming that, $g = n$, $z_1 = x_1(t)$, ..., $z_n = x_n(t)$, given a pair of $(x(t), u(t))$, the output of the fuzzy system is presented as follows:

$$y(t) = \frac{\sum_{i=1}^{m} h_i [A_i x(t) + B_i u(t)]}{\sum_{i=1}^{m} h_i}$$

$$= \sum_{i=1}^{m} \omega_i [A_i x(t) + B_i u(t)]$$

(3.1)
where

\[ h_i = \prod_{j=1}^{n} M_{ij}(x_j(t)) \]

\[ \omega_i = \frac{h_i}{\sum_{i=1}^{m} h_i} \]  \hspace{1cm} (3.2)

\( M_{ij}(x_j(t)) \) is the membership value of \( x_j(t) \) in the fuzzy set \( M_{ij} \). In general, the normalized form of \( h_i \) and \( \omega_i \) are defined as:

\[ h_i \geq 0, \ i = 1, 2, ..., m \]

\[ \sum_{i=1}^{m} h_i \geq 0 \]

\[ \omega_i \geq 0, \ i = 1, 2, ..., m \]

\[ \sum_{i=1}^{m} \omega_i = 1 \]  \hspace{1cm} (3.3)

### 3.3 Time Clustering Recognition

Time Clustering based recognition (TC) consists of two stages, model construction and hand gesture recognition as shown in Fig. 3.1 and 3.2.

#### 3.3.1 Motion Model Construction

The nonlinear functions of the finger angle trajectories are described as

\[ y_j(t) = f_j(t) \]  \hspace{1cm} (3.4)

where \( t \in \mathbb{R}^+ \) is the time instant; \( y_j(t) \in \mathbb{R}^k \) is expected angle trajectories from \( j \)th modeling motion; \( k \) is the number of finger angles of the hand; \( j = 1, 2, ..., z \); \( z \) is the total number of modeling motions; \( f \in \mathbb{R}^k \).

The equation 3.4 is linearized at the selected time points \( t_i \) as

\[ y_j(t) = y_j(t_i) + \frac{\Delta f_j(t)}{\Delta t} |_{t_i} \cdot (t - t_i) \]  \hspace{1cm} (3.5)

where \( i = 1, 2, ..., c \) is the selected time instance. A local linear equation is derived in \( t \),

\[ y_j(t) = A_{ij} \cdot t + B_{ij} \]  \hspace{1cm} (3.6)
where $A_{ij} = \frac{\Delta f_j(t_i)}{M_i} |_{t_i} \in \mathbb{R}^k$ and $B_{ij} = y_j(t_i) - \frac{\Delta f_j(t_i)}{M_i} |_{t_i} \cdot t \in \mathbb{R}^k$. According to the fuzzy modeling in equation 3.1 and 3.2, the output model from $z$ training motions is constructed by summing selected local linear models,

$$y(t) = \sum_{j=1}^{z} \sum_{i=1}^{c} \omega_{ij}(t)(A_{ij} \cdot t + B_{ij}) \quad (3.7)$$

$\omega_{ij}(t) \in [0, 1]$ is the degree of membership of the time point $t$ to the selected time instance $t_i$ from $j$th learning motion, and $\sum_{j=1}^{z} \sum_{i=1}^{c} \omega_{ij}(t) = 1$. The degree of membership is determined by

$$\omega_{ij}(t) = d_j \cdot c_i(t) \quad (3.8)$$

where $d_j \in [0, 1]$ is the weight of the $j$th training motion, and $\sum_{j=1}^{z} d_j = 1$; $c_i(t) \in [0, 1]$ is the degree of membership of the time point $t$ to the selected time instance $t_i$, and $\sum_{j=1}^{z} c_i(t) = 1$. $d_j$ is different for varied training motion according to the validity of each training dataset. For example, if the dataset is more reliable more weight will be assigned. $c_i(t)$ is calculated by

$$c_i(t) = \frac{a_i(t)}{\sum_{i=1}^{c} a_i(t)} \quad (3.9)$$

where

$$a_i(t) = \frac{1}{\sum_{j=1}^{z} \frac{(t-t_i)^T M_i(t-t_i)}{M_j(t-t_j)^T M_j(t-t_j)^T}} \quad (3.10)$$

$M_i$ is the induced matrices and $m > 1$ is the degree of fuzziness of the time instance $t_i$ contributing to the time point $t$. Equation 3.10 is from the fuzzy clustering method in (Gustafson & W.C., 1979) and equation 3.9 for normalization to ensure $\sum_{j=1}^{z} c_i(t) = 1$.

### 3.3.2 Motion Recognition

A norm function is thus proposed for hand motion recognition. Basically, given a combination of grasp sequences, the dissimilarity between the models and grasps at the time instance $t$ is

$$D(t) = ||V_{model} - V_{grasp}||_p$$

$$= \left( \sum_{i=1}^{s} |v_{model}(i) - v_{grasp}(t + i - 1)|^p \right)^{\frac{1}{p}} \quad (3.11)$$
3.4 Validation

The proposed method is evaluated by the experiment including 13 types of hand grasp motions in Fig 3.3. Each motion is repeated 10 times and all motions are from the
same subject. Half of the data is used for training the models and the rest for testing the algorithms.

Figure 3.3: Selected grasping tasks for TC

The grasp sequences are separated by a ‘intermediate state’, which in the test data is keeping the hand open and flat. The ‘intermediate state’ of the hand results in a stable dissimilarity when the model is shifted along the time series of the combined grasp sequence by equation 3.11, because the ‘intermediate state’ does not change much. The shifting and comparing are taking place in steps of time instances \( \{t, t+1, \ldots, t+s-1\} \) of the combination grasp sequence. When the model begins to overlap a grasp in the combination sequence, the dissimilarity starts to change and reaches a local minimum with a highest overlap. By calculating the dissimilarities with local minimum between models and combined grasp sequences, the most similar grasp in the combination to the model is identified with the smallest local minimum. Fig. 3.2 shows the dissimilarities when shifting five grasp models along the time series of a grasp sequence. The five local minima have been identified with star point and the global minimum is distinct from other minima.

In our experiment, the extended TC is applied to recognize grasps and it is assumed that there are all types of grasp models which are corresponding to all possible testing grasp samples, which ensures the existence of local minima for TC. Recognition results comparing GMM and HMM are provided in Table 3.1, the recognition rate of the TC method has achieved 100% in all collected data; the GMMs method is 89.23% and the HMMs method is 81.54%, which clearly shows that the TC method outperforms the rest two dominant methods. The correct recognition time intervals from 0 to 5 for each
3.4 Validation

Figure 3.4: An example of TC based recognition

type of grasp motions has shown the fact that the similar grasps such as the pair of grasp motions 5 and 13 and the pair of grasps 1 and 12 are hard to discriminate from each other. Therefore, applying a larger class for the similar grasps may improve the recognition rate significantly.

Table 3.1: Recognition results of the methods

<table>
<thead>
<tr>
<th>Grasp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>100%</td>
</tr>
<tr>
<td>GMMs</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>89.23%</td>
</tr>
<tr>
<td>HMMs</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>81.54%</td>
</tr>
</tbody>
</table>

The reason that the TC method outperforms the other two is that more Euclidean distance between the constructed models and the testing data are used to determine the hand grasping types, that is to say 500 calculations are used for a single comparison. However, much fewer such calculations are used for the models based on HMMs and
3.4 Validation

GMMs due to their state numbers. For instance, only 12 GMM components for ‘grasp 6’ are being considered, whilst 12 Euclidean distances between GMM centers and samples take effect for recognition. However 6 states only for HMM for ‘grasp 6’ are processed. Fig. 3.4 has shown that though the scope of the distances for correct models is in the range from 0.026 to 0.068, if no distance value is in this scope when comparing a sample with these 13 models, then it does not belong to anyone of these models.

In addition, the TC can also identify the time instant when a grasp action happens. However, the GMMs method only failed to recognize grasping motions 5 and 12, mainly due to the similarity of the grasp types, as shown in Fig. 3.3. The uncorrected recognition might be corrected by taking more Gaussian components. The HMMs based method gains the lowest recognition rate, in which almost half of the grasp types are not fully correctly recognized. Recognition rate of grasp motions 1 and 4 are actually below 50%.

![Figure 3.5: Recognition rate with different number of training data](image)

The increment speed of the proposed method’s recognition rate concerning different number of training data is further investigated as shown in Fig. 3.5. The whole data are divided into ten equal parts, some of which are used for training and others...
are for recognizing. From Fig. 3.5, TC has more recognition rate with fewer training data than other two methods, while GMM’s good performance mostly depends on the proper settings of the components. HMM’s performance rises sharply after using more than one third of training data, which confirms that, compared with the other two methods, HMMs is a statistical method based on Markov Chains and the principle is that the more the training data the better recognition rate. Hence, tradeoffs have to be taken according to individual applications with concerns on computational efficiency though the TC method is more suitable in general (Chapter 6).

3.5 Summary

Due to the dexterity and complexity of the human hand, recognizing human hand motions is asking for a solution with a fast learning process and an effective performance. In this chapter, time Clustering algorithm is proposed for recognizing dynamic human hand grasping gestures. The degree of membership of the model has been extended with both time instance weights and different learning motion weights in this chapter. The extended degree of membership has enabled the TC method to be more useful to learn repeated motions from the same subject or similar gestures from various subjects. The TC method models grasping gestures based on fuzzy clustering and Takagi-Sugino modeling and recognizes the gestures by comparing the constructed models with each grasping class period along the timepoints. In the experiment, 13 different types of grasping motion from CyberGloves are used. Using half grasp data as the training data, TC achieved 100% recognition rate which is better than those of GMMs and HMMs, 89.23% and 81.94%. The increment speed of the proposed method’s recognition rate concerning different number of training data is further investigated. Experiment results have also shown that the proposed TC method outperforms the HMM and GMM, two dominant recognition methods, in terms of the recognition rate.
Chapter 4

Fuzzy Gaussian Mixture Models

4.1 Introduction

As one of the most statistically mature methods for clustering (Bilik et al., 2006; David et al., 2008; Kim & Kang, 2007; Siu et al., 2009; Song et al., 2008), Gaussian Mixture Models (GMMs) are also used intensively in object tracking (Wang et al., 2008a; Wang & Yuan, 2004), background subtraction (Persson et al., 2008; Wilson et al., 2008), feature selection (Vondra & Vich, 2009; Wahab et al., 2009), signal analysis (Routtenberg & Tabrikian, 2009; Takahashi & Tsukiyama, 2009) and learning and modelling (Allali et al., 2007; Ba & Odobez, 2009; Hennebert et al., 2007; Ju et al., 2009; Montagnuolo & Messina, 2007). Various kinds of GMMs based methods are developed for specific applications, such as Adapted GMMs (Malegaonkar et al., 2007), which are used for dealing with undesired effects of variations in speech characteristics, Mahalanobis Distance based GMMs (Ververidis & Kotropoulos, 2008), which are capable of splitting one component into two new components, and Wrapped Gaussian Mixture Models (Agiomyrgiannakis & Stylianou, 2009) using an expectation-maximization algorithm suitable for circular vector data to model dispersion phases. However, more components are required when fitting the datasets with non-linear manifolds because of the intrinsic linearity of Gaussian model which leads to relative large fitting error. To solve this problem and approximate datasets with curve manifolds better, Zhang et al. (2005) proposed active curve axis Gaussian Mixture Models (AcaGMMs) which are non-linear probability models. Principle Component Analysis (PCA) and least-squares
fitting methods are used to ‘bend’ AcaGMMs in the principal plane and points are considered by handling the projection points on the principal axis. Thus AcaGMMs could be used to model datasets with the non-linear manifolds.

Fuzzy C-means (FCMs), also known as Fuzzy ISODATA, was developed by Dunn in 1973 (Dunn, 1973) and improved by Bezdek in 1981 (Bezdek, 1981). It is a popular and effective clustering method which employs fuzzy partitioning such that a data point can belong to all groups with different membership grades between 0 and 1. It employs a weighting exponent \( m \) on each fuzzy membership and distances between points and centres. The effects of the weighting exponent are discussed that optimal \( m \) may result in better performance or fast convergence in (Pal & Bezdek, 1995; Yu et al., 2004), and approaches of determining the weighting exponent have also been presented in (Choe & Jordan, 1992; Pal & Bezdek, 1995; Pei et al., 2001). The algorithmic frameworks of FCM and GMMs are closely related (Gath & Geva, 1989; Miyamoto & Mukaidono, 1997). Based on FCM, Gustafson & Kessel (1978) defined fuzzy covariance matrices of clusters which means that different clusters in the same dataset may have different geometric shapes. To define these different geometric shapes of clusters, Tran et al. (1998) further made a modification of GMMs for speaker recognition, which refined the distances in the FCM functions as the negative of logarithms of density functions. Therefore, the relationship between the membership and distance is transferred from exponential relationship to linear relationship, which however misuses exponential distance parameter to formulate Gaussian density function. Hathaway (1986) gave a general interpretation that the Expectation-Maximization (EM) algorithm of GMMs is a penalized version of the hard means clustering algorithm. Ichihashi et al. (2001) proposed a modified version of FCM with regularization by K-L information, which is similar to the EM algorithm for GMMs.

In order to combine GMMs and FCM in the mathematical modelling which is capable of fitting datasets with a fast convergence, in this chapter, conventional GMMs are firstly generalized in such a way that generalized Gaussian model is equipped with non-linearity, and then Fuzzy Gaussian Mixture Models (FGMMs) are proposed based on the generalized GMMs for a much faster convergence process (Section 4.3). The dissimilarity function in FCM maintaining the exponential relationship between membership and distance is refined for FGMMs with a degree of fuzziness in terms of the membership grades. Therefore, FGMMs not only possess non-linearity but also have
4.2 Generalized Gaussian Mixture Models

In this section, we extend the conventional GMMs into a generalized version which enables the GMMs to have capability of modelling curve datasets. Following a brief review of conventional GMMs, an EM algorithm is proposed for the generalized GMMs.

4.2.1 Conventional Gaussian Mixture Models

The probability density function for a Gaussian distribution is given by the formula (Bilmes, 1998a):

\[
p(x|\theta) = \frac{1}{\sqrt{2\pi}^d \sqrt{|\Sigma|}} \exp \left( -\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2} \right)
\]

where the set of parameters has \(\theta = (\mu, \Sigma)\), \(\mu\) is the mean, \(\Sigma\) is the covariance matrix of the Gaussian, \(d\) is the dimension of vector \(x\), and \(\exp\) denotes the exponential function.

Let \(\mathcal{X} = \{x_1, \ldots, x_n\}\) be a \(d\)-dimensional observed dataset of \(n\) vectors. If the distribution of \(\mathcal{X}\) can be modelled by a mixture of \(k\) Gaussians, the density of each vector is:

\[
p(x_t|\Theta) = \sum_{i=1}^{k} \alpha_i p_i(x_t|\theta_i)
\]

where the parameters are \(\Theta = (\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k)\) and \((\alpha_1, \ldots, \alpha_k)\) are the \(k\) mixing coefficients of the \(k\) mixed components such that \(\sum_{i=1}^{k} \alpha_i = 1\); each \(p_i\) is a
density function parameterized by \( \theta_i \). The resulting density for the samples is

\[
p(\mathcal{X}|\Theta) = \prod_{t=1}^{n} p(x_t|\Theta) = \mathcal{L}(\Theta|\mathcal{X})
\] (4.3)

The function \( \mathcal{L}(\Theta|\mathcal{X}) \) is called the likelihood of the parameters given the data, or the likelihood function. The likelihood is considered as a function of the parameters \( \Theta \) where the data \( \mathcal{X} \) is fixed. In the maximum likelihood problem, the objective is to estimate the parameter set \( \Theta \) that maximizes \( \mathcal{L} \). That is to find \( \Theta^* \) where

\[
\Theta^* = \arg \max_{\Theta} \mathcal{L}(\Theta|\mathcal{X})
\] (4.4)

Usually, the \( \log(\mathcal{L}(\Theta|\mathcal{X})) \) is maximized instead because it is analytically easier. The log-likelihood expression is given by:

\[
\log(\mathcal{L}(\Theta|\mathcal{X})) = \log \left( \prod_{t=1}^{n} p(x_t|\Theta) \right)
= \sum_{t=1}^{n} \log \left( \sum_{i=1}^{k} \alpha_i p_i(x_t|\theta_i) \right)
\] (4.5)

Directly maximizing the log-likelihood is difficult, hence an auxiliary objective function \( Q \) is taken into account:

\[
Q = \sum_{l=1}^{n} \sum_{i=1}^{k} w_{il} \log[\alpha_i p_i(x_t|\theta_i)]
\] (4.6)

where \( w_{il} \) is a posteriori probability for individual class \( i, i = 1, \ldots, k \), and it satisfies

\[
w_{il} = \frac{\alpha_i p_i(x|\theta_i)}{\sum_{s=1}^{k} \alpha_s p_s(x|\theta_s)}
\] (4.7)

and

\[
\sum_{i=1}^{k} w_{il} = 1
\] (4.8)

Maximizing equation 4.6 guarantees that \( p(\mathcal{X}|\Theta) \) is maximized if it is performed by an EM algorithm (e.g., (Bilmes, 1998b; Huang et al., 1990)). The iteration of an EM algorithm estimating the new parameters in terms of the old parameters is given as follows:

- **E-step:** compute “expected” classes of all data points for each class using equation 4.7.
4.2 Generalized Gaussian Mixture Models

- M-step: compute maximum likelihood given the data’s class membership distributions according to equations 4.9-4.11.

\[ \alpha_i^{\text{new}} = \frac{1}{n} \sum_{t=1}^{n} w_{it} \tag{4.9} \]

\[ \mu_i^{\text{new}} = \frac{\sum_{t=1}^{n} w_{it} x_t}{\sum_{t=1}^{n} w_{it}} \tag{4.10} \]

\[ \Sigma_i^{\text{new}} = \frac{\sum_{t=1}^{n} w_{it} (x_t - \mu_i^{\text{new}})(x_t - \mu_i^{\text{new}})^T}{\sum_{t=1}^{n} w_{it}} \tag{4.11} \]

When training GMMs, k-means is employed for initialization before EM starts. The iteration of EM algorithm stops when the change value of log-likelihood is below a preset threshold.

4.2.2 Generalized Gaussian Models

The conventional Gaussian model has intrinsic linearity as its axes are all beelines, so more components are needed when fitting datasets with non-linear manifolds. Active curve axis Gaussian model (AcaG) has bent principal axis, which makes it powerful in modelling curve datasets (Zhang et al., 2005).

In this section, the generalized Gaussian model is defined as the model including two modalities: one is the conventional Gaussian model with linear axes and the other is bent Gaussian or AcaG model with curve principal axis. Let \( \mathcal{X} = \{x_1, \ldots, x_n\} \) be a d-dimensional observed dataset of \( n \) vectors. The distribution of \( \mathcal{X} \) is based on one Gaussian or bent Gaussian.

First, the samples are mapped to the coordinate system determined by PCA:

\[ y_i = Q \times (x_i - T) \tag{4.12} \]

where \( T \) and \( Q \) denote the translation vector and rotation matrix of PCA individually. Let \( \mathcal{Y} = \{y_1, \ldots, y_n\} \) be the transferred coordinates of \( \mathcal{X} \) by PCA. The \( d \) dimensions of \( \mathcal{Y} \) are then denoted as \( v = [v_1, \ldots, v_d]^T \).
Then least-squares fitting method (Björck & Björck, 1996) is used to fit the co-ordinates \([v_1, v_2]^T\), which are in \(\mathcal{Y}\)'s principal plane, with the standard curve which is defined as

\[
y = ax^2 + b
\]

(4.13)

The quadratic term coefficient \(a\) indicates the bend degree of the curve. The bigger \(a\) is, the more the principal axis bends. \(b\) is the offset of the curve. After fitted by the least-squares fitting method, the parameters \(a\) and \(b\) will be determined. According to the value of parameter \(a\), the principal axis for generalized Gaussian model is defined as

\[
y = \begin{cases} 
b & (|a| < \varepsilon) 
ax^2 + b & (|a| \geq \varepsilon)
\end{cases}
\]

(4.14)

If \(|a| < \varepsilon\) where \(\varepsilon\) is a preset small positive real number, the principal axis can be considered that \(y = b\) which means the principal axis is beeline, and the generalized Gaussian model retrogresses to be conventional Gaussian model. If \(|a| \geq \varepsilon\), the principal axis is bent to be \(y = ax^2 + b\) and generalized Gaussian model with the bent axis is AcaG model. An example of generalized Gaussian model is given on the left of Fig. 4.1.

When the principal axis is beeline (i.e., \(|a| < \varepsilon\)), the probability density function of point \(x_t\) would be \(p(x_t|\theta)\) in equation 4.2.

When the principal axis is bent (i.e., \(|a| \geq \varepsilon\)), the projective points of one sample can be thought as the points on the principal axis who have the local minimal distances with the sample. The distances between sample and points along the principal axis have zero derivatives at the sample’s projective points. Given any sample \([v_{1t}, v_{2t}]^T\) which is the transferred coordinates from \(x_t\), its distance to a point \(z = [z_1, z_2]^T\) on the standard curve satisfies

\[
d^2 = (v_{1t} - z_1)^2 + (v_{2t} - z_2)^2
\]

Using the Lagrange method, the partial derivatives to \(z_1\) and \(z_2\) of \(d^2 - \lambda(z_2 - az_1^2 - b)\) are equal to zero,

\[
(z_1 - v_{1t}) - \lambda az_1 = 0
\]

\[
2(z_2 - v_{2t}) - \lambda = 0
\]

Eliminate \(\lambda\),

\[
z_1 - v_{1t} = 2az_1(z_2 - v_{2t})
\]

(4.15)
with the standard curve equation $z_2 = az_1^2 + b$, we can solve all the real roots of this equation, which are the projective points of sample $[v_{1t}, v_{2t}]$. In the principal plane, one point may have up to 3 projection points on the principal curve axis, because bending the principal axis may cause one point locates in several different normal directions of the curve axis as shown in Fig. 4.1 (B). Supposing $z = [z_1j, z_2j]^T$ is the $j$th projective point of the sample $[v_{1t}, v_{2t}]$, the arc length of $z$ is formulated as

$$l_j(v_{1t}) = \int_{(0,b)} \sqrt{(dz_1j)^2 + (dz_2j)^2}$$

$z$ is on the standard curve, so it satisfies $z_2 = az_1^2 + b$. So we have

$$l_j(v_{1t}) = \frac{1}{2}z_1j\sqrt{1 + 4a^2z_1j^2} + \frac{1}{4|a|}\ln\left(2|a|z_1j + \sqrt{1 + 4a^2z_1j^2}\right) \quad (4.16)$$

The distance between the sample $[v_{1t}, v_{2t}]^T$ and its projective point $z$ is

$$l_j(v_{2t}) = \sqrt{(v_{1t} - z_1j)^2 + (v_{2t} - z_2j)^2} \quad (4.17)$$

Then the probability density function of point $x_t$ is the sum probability of its projective points:

$$p(x_t|\theta) = \sum_{j=1}^{J_t} \prod_{s=1}^{2} \frac{\exp(-l_j^2(v_{st})/2\Sigma_s)}{\sqrt{2\pi|\Sigma_s|}} \prod_{s=3}^{d} \frac{\exp(-v_{st}^2/2\Sigma_s)}{\sqrt{2\pi|\Sigma_s|}} \quad (4.18)$$
4.2 Generalized Gaussian Mixture Models

Where the parameter set \( \theta = (\mu, \Sigma, C, Q, T) \); \( C \) denotes the control parameters of a standard curve \( f : y = ax^2 + b \); \( J_t \) is the number of the projective points of sample \( x_t \).

4.2.3 Modified Expectation Maximisation Algorithm

Generalized GMMs combine conventional GMMs with AcaGMMs, both of whom employ EM algorithm for parameters estimation. With the renewed probability density function 4.18, the iteration of EM algorithm for generalized GMMs is proposed:

• E-step: compute “expected” classes of all data points for each class. If \(|a| < \varepsilon\), \( p(x_t|\theta_i) \) is computed by equation 4.1; otherwise, equation 4.18 is employed. Then \( w_{it} \) is calculated by equation 4.7.

• M-step: compute maximum likelihood given the data’s class membership distributions. \( \alpha_{ti}^{new} \) is estimated from equation 4.9, then the following are obtained,

\[
\chi_i' = [w_{i1}, \ldots, w_{in}] \cdot [x_1, \ldots, x_n] \quad (4.19)
\]

\[
(C_i^{new}, T_i^{new}, Q_i^{new}) = LSFM(PCA(\chi_i')) \quad (4.20)
\]

\( PCA() \) is PCA function for estimating the translation matrix \( T_i^{new} \) and rotation matrix \( Q_i^{new} \). \( LSFM() \) is Least-Squares Fitting Method for estimating control parameters \( C_i^{new} = (a, c) \) which shapes the curve axis with standard curve \( y = ax^2 + b \); If \(|a| < \varepsilon\); \( \mu_{ti}^{new} \) and \( \Sigma_{ti}^{new} \) are assessed by equations 4.10 and 4.11; otherwise, they can be computed by the following equations.

\[
\mu_{ti}^{new} = \sum_{t=1}^{n} \frac{w_{it}x_t}{\sum_{t=1}^{n} w_{it}} + (Q_i^{new})^{-1} \begin{bmatrix} 0, b, 0, \ldots, 0 \end{bmatrix}^T + T_i^{new} \quad (4.21)
\]

\[
\bar{L}_{te}^{(i)} = \frac{\sum_{j=1}^{J_t} p(l_j^{(i)}(v_{et})|N(0, \Sigma_{te}))}{\sum_{j=1}^{J_t} p(l_j^{(i)}(v_{et})|N(0, \Sigma_{te}))} \quad (e = 1, 2) \quad (4.22)
\]
4.3 Fuzzy Gaussian Mixture Models

\[
\Sigma_{ie}^{\text{new}} = \frac{\sum_{t=1}^{n} w_{it} \bar{L}_{t}^{(i)}}{\sum_{t=1}^{n} w_{it}} \quad (e = 1, 2) \tag{4.23}
\]

\[
\Sigma_{i(3-d)}^{\text{new}} = \frac{\sum_{t=1}^{n} w_{it}(x_{t} - \mu_{i}^{\text{new}})_{(3-d)}(x_{t} - \mu_{i}^{\text{new}})_{(3-d)}^{T}}{\sum_{t=1}^{n} w_{it}} \tag{4.24}
\]

If that sample point \(x_{t}\) has \(J_{t}^{(i)}\) projective points on the \(i\)th component, the arc length of the \(j\)th projective point is \(l_{j}^{(i)}(v_{1i})\) and the distance between \(j\)th projective point and \(x_{t}\) is \(l_{j}^{(i)}(v_{2i})\). \(\bar{L}_{i}^{(i)}\) is the average arc length of \(x_{t}\)'s projective points on the \(i\)th component, and \(\bar{L}_{i}^{(i)}\) is the average distance between \(x_{t}\) and its projective points on the \(i\)th component. \(p(x|N(\mu, \Sigma))\) is the probability density of \(x\) in the Gaussian with \(\mu\) and \(\Sigma\). \(\Sigma_{i1}\) and \(\Sigma_{i2}\) denote the two covariances in the \(i\)th principal axis, and \(\Sigma_{i(3-d)}\) is the covariance in the \(d-3\) dimension space orthogonal to the principal plane.

When \(|a|\) is very small in generalized GMMs, AcaGMM can be converted to be GMM so that \(\mu_{i}^{\text{new}}\) and \(\Sigma_{i}^{\text{new}}\) will be assessed only by equation 4.10 and 4.11 instead of equations 4.21-4.24, which would save the computational cost of AcaGMM. The switching equation 14 does not play a role in improving the fitting degree of AcaGMM, but it saves unnecessary steps to estimate the new centres and covariances when \(|a|\) < \(\varepsilon\). So that the proposed generalized GMMs are faster than the AcaGMMs especially when the principal axis of any component has a small degree of curvature.

Like conventional GMMs, the generalized GMMs are also trained by EM algorithm with k-means clustering initialization; the number of generalized GMMs components in any dataset also needs to be selected properly. Thanks to its non-linear fitting, the components of the generalized GMMs is fewer than those of the conventional GMMs. An example is provided in Fig. 4.2. The iteration does not stop until the relative difference of the log-likelihood obtained by equation 4.5 reduces under the preset threshold.

4.3 Fuzzy Gaussian Mixture Models

Inspired from the mechanism of FCM, the introduction of the weighting exponent on the fuzzy membership into generalized GMMs is proposed in order to improve the
efficiency of convergence. It is evident that GMMs are mixtures of Gaussian distributions, the dissimilarities of individual cluster points are defined in form of certain exponential function of the distances. Two types of fuzzy generalized GMMs are proposed based on different dissimilarity functions denoted by the exponential distance, i.e., probability based FGMMs and distance based FGMMs.

### 4.3.1 Fuzzy C-means Clustering

FCM is a method of clustering which allows one piece of data to belong to two or more clusters, which is frequently used in pattern recognition (Bezdek, 1981). Let \( X = \{ x_1, x_2, \ldots, x_n \} \) be the \( d \) dimensional observed dataset with \( n \) vectors, \( k \) be the number of clusters with \( 2 \leq k \leq n \), \( m \) be a weighting exponent on each fuzzy membership, and the degree of fuzziness, \( \mu_i \), be the prototype of the centre of \( i \)th cluster, and \( U = \{ u_{it} \} \) where \( u_{it} \) is the degree of membership of \( x_t \) in the \( i \)th cluster and has

\[
0 \leq u_{it} \leq 1; \sum_{i=1}^{k} u_{it} = 1 \tag{4.25}
\]
1 ≤ i ≤ k and 1 ≤ t ≤ n. The dissimilarity function with an $A$ norm distance measure between object $x_t$ and cluster centre $\mu_i$ is

$$d_{it}^2 = ||x_t - \mu_i||_A^2 = (x_t - \mu_i)^T A(x_t - \mu_i)$$ (4.26)

The aim of FCM is to find the new cluster centres (centroids) that minimize a dissimilarity function, i.e. the weighted within group sum of squared error objective function $J_m(U, \mu, X)$:

$$J_m(U, \mu, X) = \sum_{t=1}^{n} \sum_{i=1}^{k} (u_{it}^m d_{it}^2)$$ (4.27)

The process of minimizing object function $J_m$ depends on how centres find their ways to the best positions, as the fuzzy memberships $u_{it}$ and norm distance $d_{it}$ would change along with the new centres’ position. Given the previous positions of the centres, equation 4.26 would produce distance based dissimilarities. To minimize the object function 4.27, new positions of centres would be determined by the following equations:

$$u_{it} = \left[ \sum_{j=1}^{k} \left( \frac{d_{it}}{d_{jt}} \right)^{\frac{2}{m-1}} \right]^{-1}$$ (4.28)

$$\mu_{i}^{new} = \frac{\sum_{t=1}^{n} u_{it}^m x_t}{\sum_{t=1}^{n} u_{it}^m}$$ (4.29)

where $\mu_{i}^{new}$ is the new position of $i$th centre estimated by the degree of membership with the weighting exponent $m$.

After a number of iterations using equations 4.26-4.29, the centres of clusters are optimized to minimize the object function. The iteration stops when difference of the current value of object function and the previous value of object function is less than the preset threshold.

As FCM introduces a weighting exponent $m$, the degree of fuzziness, on each fuzzy membership in equations 4.27 and 4.29, the points which are nearer to one cluster than other clusters are made more important for this cluster and at the same time more insignificant for other clusters.
4.3 Fuzzy Gaussian Mixture Models

4.3.2 Probability based Fuzzy Gaussian Mixture Models

For a faster convergence speed of GMMs, a dissimilarity function for probability based FGMMs is defined as:

\[
d^2_{it} = \frac{1}{\alpha_i p_i(x_t|\theta_i)}
\]

(4.30)

where \(\alpha_i\) is the weight of \(i^{th}\) component; \(p_i(x_t|\theta_i)\) is from equation 4.1 for \(|a| < \varepsilon\) or equation 4.18 for \(|a| \geq \varepsilon\).

Compared to equation 4.26, the dissimilarity function defined by equation 4.30 consists of not only the distance but also the covariance and mixture weights, and it is in direct ratio with the exponent of the distance. The objective function is extended from equation 4.27 as:

\[
J_m(U, X, \mu, \Sigma) = \sum_{t=1}^{n} \sum_{i=1}^{k} (u_{mit}^m d^2_{it})
\]

(4.31)

Minimizing \(J_m\) is performed by the same \(u_{it}\) operations of FCM using equations 4.28. The new mixture weights are estimated by equation 4.32:

\[
\alpha_{i}^{new} = \frac{\sum_{t=1}^{n} u_{mit}^m}{\sum_{i=1}^{k} \sum_{t=1}^{n} u_{it}^m}
\]

(4.32)

The parameters of curve axis \((a_i, b_i)\) in \(C_i\) can be achieved from equation 4.20. If \(|a_i| < \varepsilon\) (i.e., the principal axis of component \(i\) is beeline) then the generalized Gaussian model regresses to be a conventional Gaussian model, the centre and the covariance can be obtained by equations 4.29 and 4.33.

\[
\Sigma_{i}^{new} = \frac{\sum_{t=1}^{n} u_{it}^m (x_t - \bar{\mu}_i)(x_t - \bar{\mu}_i)^T}{\sum_{t=1}^{n} u_{it}^m}
\]

(4.33)

Otherwise when \(|a_i| \geq \varepsilon\), the principal axis of component \(i\) would be shaped as standard curve \(y = a_i x^2 + b_i\) and the generalized Gaussian model turns into an AcaG model. The parameters of standard curve (translation and rotation matrices), centres and covariances can be computed through equations 4.20, and 4.34-4.36.

\[
\mu_{i}^{new} = \frac{\sum_{t=1}^{n} u_{it}^m x_t}{\sum_{t=1}^{n} u_{it}^m} + (Q_{i}^{new})^{-1} \underbrace{[0, b_i, 0, \ldots, 0]}_{d}^T + T_{i}^{new}
\]

(4.34)
4.3 Fuzzy Gaussian Mixture Models

\[ \Sigma_{\text{new}}^{\text{new}} = \frac{\sum_{i=1}^{n} u_{im}^{e} L_{i}^{e}}{\sum_{i=1}^{n} u_{im}^{e}} \quad (e = 1, 2) \]  (4.35)

\[ \Sigma_{\text{new}}^{i(3-d)} = \frac{\sum_{i=1}^{n} u_{im}^{e}(x_{t} - \mu_{i})^{(3-d)}(x_{t} - \mu_{i})^{T}(3-d)}{\sum_{i=1}^{n} u_{im}^{e}} \]  (4.36)

The EM algorithm of probability based FGMMs is provided in algorithm 1 where \( d_{it} \) is obtained from equation 4.30.

### 4.3.3 Distance based Fuzzy Gaussian Mixture Models

Referring to equation 4.30, the degree of fuzziness in probability based FGMMs takes effect on the distance between any point and its centre point, and the mixture weights. Alternatively, another solution is this dissimilarity function which focuses the effect of degree of fuzziness only on the distances between points and their component centres.

When the principal axis is bent, the distances in principal plane consists of arc length and normal length, equation 4.18 can be rewritten as,

\[ p(x_{t} | \theta) = \left( \frac{f_{t}}{\sum_{j=1}^{n} \prod_{s=1}^{2} \exp \left( \frac{-x_{t} | \theta_{j} |^{m}}{2 \alpha_{m} (m-1)} \right)} \prod_{s=3}^{d} \exp \left( \frac{-x_{t} | \theta_{j} |^{m}}{2 \alpha_{m} (m-1)} \right) \right)^{\frac{m-1}{m}} \]  (4.37)

Furthermore, based on above equation the dissimilarity function can be defined as follows,

\[ d_{it}^{2} = \begin{cases} \exp \left( \frac{(x_{t} - \mu_{i})^{T} \Sigma_{i}^{-1} (x_{t} - \mu_{i})}{\alpha_{m} (2\pi)^{\frac{1}{2}} | \Sigma_{i} |^{-\frac{1}{2}} \frac{m-1}{m}} \right) & (|a_{i}| < \varepsilon) \\ \frac{1}{\alpha_{m} - m} p_{i}(x_{t} | \theta_{i}) & (|a_{i}| \geq \varepsilon) \end{cases} \]  (4.38)

where \( m \) is the degree of fuzziness; \( p_{i}(x_{t} | \theta_{i}) \) is obtained from equation 4.37. This dissimilarity function is in the form of direct a ratio with the exponent of the distance. Its objective function, membership function and the functions estimating centre vectors, mixture weights and covariances are the same as probability based FGMMs, referring to equations 4.34-4.36. The EM algorithm for distance based FGMMs is the
4.3 Fuzzy Gaussian Mixture Models

Algorithm 1 EM algorithm of probability based FGMMs.

Require: Fix $k$, $m$ and $\varepsilon \{ k$ is the number of components $2 < k < n$; $m$ is the degree of fuzziness $m > 1$; $\varepsilon$ is a small preset real positive number\}.

1: $U \leftarrow fcm(data, k)$ \{Use FCM to generate matrix $U$ satisfying equation 4.25\}
2: repeat
3: \hspace{1em} for all $i$ such that $1 \leq i \leq k$ do
4: \hspace{2em} $\alpha_i^{new} \leftarrow equation$ \{Compute the $k$ fuzzy mixture weights $\alpha_i^{new}$ using equation 4.32\}
5: \hspace{2em} $a_i \leftarrow equation$
6: \hspace{2em} if $a_i < \varepsilon$ then
7: \hspace{3em} $\mu_i^{new} \leftarrow equation$ \{Compute the $k$ fuzzy mean vectors using equation 4.29\}
8: \hspace{3em} $\Sigma_i^{new} \leftarrow equation$ \{Compute the $k$ fuzzy covariance matrices using equation 4.33\}
9: \hspace{2em} else
10: \hspace{3em} \{ $C_i^{new}, T_i^{new}, Q_i^{new}$ \} $\leftarrow equation$ \{Compute the $k$ fuzzy control parameters of standard curve, translation and rotation matrices using equation 4.20\}
11: \hspace{3em} $\mu_i^{new} \leftarrow equation$ \{Compute the $k$ fuzzy centers using equation 4.34\}
12: \hspace{3em} $\Sigma_i^{new} \leftarrow equations$ \{Compute the $k$ fuzzy covariance matrices using equations 4.35-4.36\}
13: \hspace{2em} end if
14: \hspace{1em} end for
15: $U^{new} \leftarrow equation$ \{Upgrade the fuzzy membership using equation 4.28\}
16: $\log(L(\Theta|\mathcal{X})^{new} \leftarrow equation$ \{Compute the log-likelihood using equation 4.5\}
17: until $\frac{\log(L(\Theta|\mathcal{X})^{new}}{\log(L(\Theta|\mathcal{X})^{old}} - 1 \leq threshold$ \{Stop if the relative difference of the log-likelihood between two adjacent iterations is blow the preset threshold\}
same as probability based FGMMs shown in algorithm 1 except that \( d_{it} \) is provided by equation 4.38.

### 4.3.4 Comparison of Probability/Distance based Fuzzy Gaussian Mixture Models

To clarify the difference between probability based FGMMs and distance based FGMMs, \( u_{it}^m \) can be obtained by their different dissimilarity functions. From equations 4.28 and 4.30, \( u_{it}^m \) in probability based FGMMs can be obtained when \( |a| < \varepsilon \):

\[
u_{it}^m = \frac{\left[ \alpha_i p_i(x_t | \theta_i) \right]^{m-1}}{\left[ \sum_{j=1}^{k} \left( \alpha_j p_j(x_t | \theta_j) \right)^{m-1} \right]^{1 / m}}
\]

so that

\[
u_{it}^m \propto \left[ \alpha_i p_i(x_t | \theta_i) \right]^{m-1} \tag{4.40}
\]

which illustrates that the \( u_{it}^m \) is in direct ratio with the fuzzy probability.

In contrast, equation 4.38 is substituted to equation 4.28 and then \( u_{it}^m \) can be obtained for distance based FGMMs under condition of \( |a| < \varepsilon \),

\[
u_{it}^m = \frac{\alpha_i (2\pi)^{-d/2} |\Sigma_i|^{-1/2} \exp \left( \frac{-1}{2} \frac{(x_t - \mu_i)^T \Sigma_i^{-1} (x_t - \mu_i)}{m-1} \right)}{\left[ \sum_{j=1}^{k} \alpha_j (2\pi)^{-d/2} |\Sigma_j|^{-1/2} \exp \left( \frac{-1}{2} \frac{(x_t - \mu_j)^T \Sigma_j^{-1} (x_t - \mu_j)}{m-1} \right) \right]^{1 / m}}
\]

so that

\[
u_{it}^m \propto \exp \left( \frac{-1}{m-1} \frac{(x_t - \mu_i)^T \Sigma_i^{-1} (x_t - \mu_i)}{2} \right) \tag{4.42}
\]

Different from probability based FGMMs, the membership function of distance based FGMMs is directly proportional to the exponent of the fuzzy distances. Similar comparison can be achieved for \( |a| \geq \varepsilon \), so it is the reason that the former is named as probability based FGMMs and the later is called distance based FGMMs.

The distance based FGMMs are similar with FCM with K-L information term (KLFCM) (Ichihashi et al., 2001) since each adds a parameter on the distance exponent of the fuzzy membership \( u_{it} \). The weighting exponent ‘\( m \)’ in distance based FGMMs and the parameter ‘\( \lambda \)’ in KLFCM are used to control the relative distances between points to avoid the influence of noise data and to derive a faster convergence. KLFCM is considered to be a FCM-type counterpart of GMMs proposed by Ichihashi et al., who
4.3 Fuzzy Gaussian Mixture Models

generalized the regularized objective function replacing the entropy term (Miyamoto & Mukaidono, 1997) with K-L information term (Honda & Ichihashi, 2005). Hence, KLFBCM is based on the minimization of the modified objective function from entropy regularization, while the distance based FGMMs is aimed at maximizing the likelihood function, which will be proved below. In addition, although ‘λ’ in KLFBCM and ‘m’ in distance based FGMMs all play a role in tuning the degree of fuzziness, the weighting exponent ‘m’ in distance based FGMMs has the same meaning/effects as in FCM (Bezdek, 1981; Cannon et al., 1986; Pal & Bezdek, 1995; Yu et al., 2004), so the methods of determining the weighting exponent in FCM (Choe & Jordan, 1992; Pal & Bezdek, 1995; Pei et al., 2001) for better performances may also be used directly in distance based FGMMs.

4.3.5 Mathematical Proof

The proposed updating algorithm not only minimizes the objective function 4.27 of FCM but also maximizes the log-likelihood function 4.5 of GMMs. The theoretical justification is presented as below.

- Maximizing the log-likelihood function 4.5 of GMMs

The EM algorithm of normal GMMs guarantees the maximum likelihood function 4.5. In practice, the objective $Q$ function 4.6 is taken instead.

Comparing the two EM algorithms of normal GMMs (equations 4.7, 4.9, 4.10, 4.11) and Fuzzy GMMs ($|a| < \varepsilon$) (equations 4.28, 4.29, 4.32, 4.33), the only difference is that $u_{im}^m$ is used as $w_{im}$ in Fuzzy GMMs. Therefore, the EM algorithm of Fuzzy GMMs ($|a| < \varepsilon$) would guarantee the objective $Q^*$ function:

$$Q^* = \sum_{t=1}^{n} \sum_{i=1}^{k} u_{im}^m \log[\alpha_i p_i(x_t|\theta_i)] \tag{4.43}$$

In Section 4.3.4, it is shown that for probability based FGMMs

$$u_{im}^m \propto [\alpha_i p_i(x_t|\theta_i)]^{m-1} \tag{4.44}$$

and for distance based FGMMs

$$u_{im}^m \propto \exp\left(\frac{-m^{-1} (x_t-\mu_i)^T \Sigma^{-1}_{i}(x_t-\mu_i)}{2}\right) \tag{4.45}$$
4.4 Validation

Because

\[ w_{it} = \frac{\alpha_ip_i(x_t|\theta_i)}{\sum_{s=1}^{k} \alpha_sp_s(x_t|\theta_s)} \] (4.46)

for a fixed \( m \), there exists a function \( f \) that \( u_{it}^m = f(w_{it}) \), which is a monotonic increasing function. So that there exists a function \( g \) that \( Q^* = g(Q) \), which is also a monotonic increasing function. When \( Q^* \) reaches to its local maximum, \( Q \) also gets local maximum. Therefore, the EM algorithm of FGMM (\(|a| < \varepsilon\)) guarantees a local optimal search.

The same result can be achieved for the EM algorithm of FGMM (\(|a| \geq \varepsilon\)) by a similar justification.

- **Minimizing the objection function 4.27 of FCM**

  Because the degree of membership \( u_{it} \) and the position of centre \( \mu_i \) in our proposed EM algorithm are achieved by the same way as in FCM, i.e. by the functions 4.28 and 4.29, the objection function 4.27 will be minimized.

4.4 Validation

The proposed FGMMs are compared with both conventional and generalized GMMs, respectively, on Gaussian based datasets and analysis of written characters for demonstrating the individual performance. Since initialization of EM algorithm may cause different results of the same settings, in order to show the effectiveness of the fuzzy algorithms in fairness, the initializations of all methods are implemented by FCM and each comparison has only one execution of FCM, which would trigger running of EM algorithms. For example, when comparing generalized GMMs with probability based FGMMs and distance based FGMMs, all of these three methods are initialized by FCM which is implemented once at the beginning of comparison and the membership grades that results are used by all the three methods simultaneously. In addition, in order to have a fair comparison of conventional GMMs and the FGMMs, \( a \) is set as 0 for all components in probability based FGMMs and distance based FGMMs in the following experiments so that all the compared methods have elliptical components. For comparison of generalized GMMs and the FGMMs, the value of \( a \) is calculated from equation 4.20 where \(|a| \geq 0\).
4.4 Validation

4.4.1 Gaussian based Datasets

Figure 4.3: Test data set: Data of five conventional Gaussian mixtures (left) and data of five generalized Gaussian mixtures (right).

1000 2-dimensional points of 5 conventional Gaussian mixtures are generated for comparisons of conventional GMMs, probability based FGMMs and distance based FGMMs, as shown on the left of Fig. 4.3, and each mixture has 200 points. 1600 of 2-dimensional points are created according to the proposed theory of generalized Gaussian model in Section 4.2.2 for the comparisons, and the dataset has 5 mixtures as well on the right of Fig. 4.3.

First, the number of components is set as 5 to test the performance of the proposed methods with the suitable number of components. Structures of the most datasets are, however, usually unknown beforehand and obtaining general ideas of the structures would be useful for further data analysis. Hence, different numbers of components are considered in comparison with the 5-component constructed datasets in order to achieve a more comprehensive understanding of data structure estimation.
Figure 4.4: Results of conventional GMMs (56 iterations, 0.257 second, left), probability based FGMMs ($\alpha = 0$) (48 iterations, 0.235 second, middle) and distance based FGMMs ($\alpha = 0$) (35 iterations, 0.232 second, right) with 5 components on the dataset of five conventional Gaussian mixtures.

### 4.4.1.1 Five Components Gaussian

Five components are set for conventional GMMs, probability based FGMMs and distance based FGMMs, the results can be seen in Fig. 4.4, where the locations of the five components of these three methods are similar.

Their convergence processes are shown in Fig. 4.5 in which the log-likelihoods tend to smooth-out with the iterations of EM algorithm. Numbers (i.e., 1-3) stand for different thresholds: ‘1’ means the relative difference of log-likelihood, calculated by algorithm 1 between current iteration and the last iteration, is less than threshold of $10^{-8}$; ‘2’ is for the threshold of $10^{-9}$ and ‘3’ is for the mostly used and the least threshold of $10^{-10}$. In Fig. 4.5, these three methods have similar capabilities to fit the dataset since values of the three stabilized log-likelihoods are -5614.5, -5615.31 and -6120.5 for conventional GMMs, probability based FGMMs and distance based FGMMs, respectively. However, the numbers of iterations of probability based FGMMs and distance based FGMMs are 48 in 0.235 second and 35 in 0.232 second which are less than the iteration number used by GMMs (i.e., 56 in 0.257 second) in terms of threshold value $10^{-10}$; the convergence speeds of probability based FGMMs and
4.4 Validation

Figure 4.5: Log-likelihoods during convergences of conventional GMMs, probability based FGMMs ($a = 0$) and distance based FGMMs ($a = 0$) EM algorithms with 5 components on the dataset of five conventional Gaussian mixtures.

Distance based FGMMs are faster than conventional GMMs from the distributions of these three thresholds.

For the data of five generalized Gaussian mixtures, Fig. 4.6 presents the fitting results of generalized GMMs, probability based FGMMs and distance based FGMMs, where little difference could be seen since the stabilized log-likelihoods locate at -2494.11, -2494.03 and -2494.09 for generalized GMMs, probability based FGMMs and distance based FGMMs individually. The distributions of the threshold markers of ‘1-3’ in the figure illustrate convergence speeds of probability based FGMMs and distance based FGMMs evidently are faster than that of generalized GMMs.

4.4.1.2 Various Components Gaussian

We further investigate the comparison of these three methods’ performances when setting the number $k$ of components different value from 3 to 7; the performance criteria
4.4 Validation

Figure 4.6: Results of generalized GMMs (11 iterations, 2.041 seconds, left), probability based FGMMs ($|a| \geq 0$) (7 iterations, 1.562 seconds, middle) and distance based FGMMs ($|a| \geq 0$) (8 iterations, 1.516 seconds, right) with 5 components on the dataset of five generalized Gaussian mixtures.

include log-likelihood and convergence speed. The convergence speed can be judged from the number of iterations and the computational time for the threshold of $10^{-10}$. In the following tables, ‘LL’ stands for log-likelihoods, ‘NI’ for number of iterations and ‘Time’ for the computational time.

Table 4.1 shows the results of NIs, LLs and time of conventional GMMs, probability based FGMMs and distance based FGMMs when setting $k$ from 3 to 7 on the data of five conventional Gaussian mixtures. The results in the table demonstrate that, with same number of components, log-likelihoods of these three methods are similar to each other, but probability based FGMMs and distance based FGMMs have much smaller numbers of iterations and cost time in each row than conventional GMMs, which means the FGMMs not only have the GMMs’ capability of fitting the data but also are capable of reducing the computational cost in comparison with GMMs. To have a general understanding of fuzzy GMMs’ capabilities of computational efficiency, we calculate the NI average percentages (‘AP’ in the table) of probability based FGMMs and distance based FGMMs relative to conventional GMMs using the following
4.4 Validation

Figure 4.7: Log-likelihoods during convergences of generalized GMMs, probability based FGMMs ($|a| \geq 0$) and distance based FGMMs ($|a| \geq 0$) EM algorithms with 5 components on the dataset of five generalized Gaussian mixtures.

The formula:

$$AP = \frac{\sum_{k=3}^{5} \frac{NI_k^F}{NI_k^G}}{5} \times 100\%$$  \hspace{1cm} (4.47)

where $NI_k^F$ denotes the number of iterations of fuzzy algorithms when setting number of component as $k$, $NI_k^G$ stands for the number of iterations of Gaussian algorithms when setting number of component as $k$. The result of AP of conventional GMMs is set as 100% and probability based FGMMs take only 63.49% of the iterations of conventional GMMs and distance based FGMMs take only 60.56% in table 4.1. The same operation is taken for the computational time, and the results are listed in the last row of the table. It shows that these two FGMMs have only 79.79% and 64.07% of the cost time of conventional GMMs.

Considering generalized GMMs, table 4.2 shows the results of NIs, LLs and time of generalized GMMs, probability based FGMMs and distance based FGMMs when setting $k$ from 3 to 7 on the data of five generalized Gaussian mixtures. Log-likelihoods
Table 4.1: Results of conventional GMMs (CGMM), probability based FGMMs (PFGMM, \( a = 0 \)) and distance based FGMMs (DFGMM, \( a = 0 \)) with 3-7 components on the dataset of five conventional Gaussian mixtures.

<table>
<thead>
<tr>
<th>k</th>
<th>CGMMs</th>
<th>PFGMMs ((a = 0))</th>
<th>DFGMMs ((a = 0))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LL</td>
<td>NI</td>
<td>Time(s)</td>
</tr>
<tr>
<td>3</td>
<td>-5857.1</td>
<td>172</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68</td>
<td>0.219</td>
</tr>
<tr>
<td>4</td>
<td>-5702.0</td>
<td>78</td>
<td>0.312</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>0.202</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48</td>
<td>0.171</td>
</tr>
<tr>
<td>5</td>
<td>-5614.5</td>
<td>56</td>
<td>0.257</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>0.232</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48</td>
<td>0.235</td>
</tr>
<tr>
<td>6</td>
<td>-5610.2</td>
<td>623</td>
<td>3.282</td>
</tr>
<tr>
<td></td>
<td></td>
<td>488</td>
<td>3.175</td>
</tr>
<tr>
<td></td>
<td></td>
<td>354</td>
<td>2.203</td>
</tr>
<tr>
<td>7</td>
<td>-5600.2</td>
<td>811</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>611</td>
<td>3.918</td>
</tr>
<tr>
<td></td>
<td></td>
<td>480</td>
<td>2.906</td>
</tr>
<tr>
<td>3-7</td>
<td>AP</td>
<td>100%</td>
<td>63.49%</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>100%</td>
<td>79.79%</td>
</tr>
</tbody>
</table>

of all the three methods are similar to each other, which proves the FGMMs have the similar capability of fitting datasets. On the other hand, the numbers of iterations and cost time of probability based FGMMs and distance based FGMMs are much less than those of generalized GMMs. The NI and time average percentages of probability based FGMMs and distance based FGMMs are computed according to equation 4.47 shown in the Table 4.2; it also demonstrates that the FGMMs are powerful in reducing the computational cost.
Table 4.2: Results of generalized GMMs (GGMMs), probability based FGMMs ($|a| \geq 0$) and distance based FGMMs ($|a| \geq 0$) with 3-7 components on the dataset of five generalized Gaussian mixtures.

|       | GGMMs | PFGMMs ($|a| \geq 0$) | DFGMMs ($|a| \geq 0$) |
|-------|-------|---------------------|---------------------|
|       | LL    | NI                  | Time(s)             |
| $k=3$ | -11435 | 85                  | 10.203              |
|       | -11430 | 48                  | 6.031               |
|       | -12217 | 44                  | 5.516               |
| $k=4$ | -10855 | 8                   | 1.313               |
|       | -10855 | 6                   | 1.047               |
|       | -12440 | 7                   | 1.234               |
| $k=5$ | -10472 | 11                  | 2.041               |
|       | -10471 | 7                   | 1.512               |
|       | -11272 | 8                   | 1.516               |
| $k=6$ | -10478 | 114                 | 18.047              |
|       | -10489 | 80                  | 19.528              |
|       | -12066 | 59                  | 13.953              |
| $k=7$ | -10479 | 191                 | 47.078              |
|       | -10495 | 95                  | 25.734              |
|       | -12080 | 84                  | 22.188              |
| $k=3-7$ | AP | 100% | 63.11% | 61.55% |
|       | Time  | 100% | 75.16% | 69.35% |

4.4.2 Structure Analysis of Characters

GMMs is one of the most dominant methods widely used in structure modelling and generalized GMMs show powerful functioning in modelling data with curve manifolds. Therefore, we choose the application for structure analysis of characters to test the effectiveness of the proposed fuzzy methods.

Fig. 4.8 shows the results of fitting character ‘R’ by conventional GMMs and the FGMMs ($a = 0$), where $k$ is set to be 9 for accepted recognition. Their log-likelihoods are -8995.3, -9047.6 and -9669.0, evidently whose difference is relatively small.

Their convergence processes are presented in Fig. 4.9, where the numbers of iter-
4.4 Validation

Figure 4.8: Results of conventional GMMs (1263 iterations, 10.75 seconds, left), probability based FGMMs ($a = 0$) (520 iterations, 6.41 seconds, middle) and distance based FGMMs ($a = 0$) (473 iterations, 4.05 seconds, right) with 9 components on the character ‘R’.

Iterations are 1263, 520 and 473 for conventional GMMs, probability based FGMMs and distance based FGMMs, respectively, in terms of the threshold of $10^{-10}$. The FGMMs take about half computational cost of conventional GMMs, since their computational time are 6.41 seconds and 4.05 seconds compared to 10.75 seconds by conventional GMMs.

Fig. 4.10 shows that the results of fitting character ‘R’ by generalized GMMs and the FGMMs ($|a| \geq 0$). It demonstrates that generalized GMMs outperforms conventional GMMs due to its capability of data fitting with curve manifolds.

Their convergence processes are presented in Fig. 4.11, where the numbers of iterations are 649, 257 and 197 for generalized GMMs, probability based FGMMs and distance based FGMMs, respectively, for the threshold of $10^{-10}$, and their computational time are 133.19, 55.00 and 40.75 seconds. Therefore, the FGMMs significantly outperforms generalized GMMs again.
Figure 4.9: Log-likelihoods during convergences of conventional GMMs, probability based FGMMs ($a = 0$) and distance based FGMMs ($a = 0$) EM algorithms with 9 components on the character ‘R’.
4.4 Validation

Figure 4.10: Results of generalized GMMs (649 iterations, 133.19 seconds, left), probability based FGMMs ($|a| \geq 0$) (257 iterations, 55 seconds, middle) and distance based FGMMs ($|a| \geq 0$) (197 iterations, 40.75 seconds, right) with 7 components on the character ‘R’.

Figure 4.11: Log-likelihoods during convergences of generalized GMMs, probability based FGMMs ($|a| \geq 0$) and distance based FGMMs ($|a| \geq 0$) EM algorithms with 7 components on the character ‘R’.
Table 4.3: Results of all models with $k$ components and $m$ degree of fuzziness on datasets of ‘6’, ‘8’, ‘a’, ‘B’ and ‘R’.

| Data | k  | m  | NI/LL | CGMMs ($a = 0$) | PFGMMs ($|a| ≥ 0$) | DFGMMs ($|a| ≥ 0$) |
|------|----|----|-------|----------------|-----------------|----------------|
| ‘6’  | 6  | 3  | LL    | -14008         | -14014          | 14932          |
|      |    |    | NI    | 571           | 385             | 394            |
|      |    |    | Time(s) | 4.106         | 3.457           | 3.078          |
| ‘8’  | 9  | 2  | LL    | -7397.3       | -7404.4         | -8450.9       |
|      |    |    | NI    | 802           | 468             | 447           |
|      |    |    | Time(s) | 6.719         | 4.563           | 3.937          |
| ‘a’  | 9  | 3  | LL    | -8816.5       | -8830.7         | -9450.6       |
|      |    |    | NI    | 6530          | 2942            | 2092          |
|      |    |    | Time(s) | 53.61         | 32.02           | 19.31          |
| ‘B’  | 10 | 4  | LL    | -8704.7       | -8709.2         | -9113.2       |
|      |    |    | NI    | 2204          | 1204            | 533           |
|      |    |    | Time(s) | 22.14         | 17.42           | 6.844          |
| ‘R’  | 9  | 3  | LL    | -8995.3       | -9047.6         | -9669.0       |
|      |    |    | NI    | 1263          | 520             | 437           |
|      |    |    | Time(s) | 10.75         | 6.01            | 4.047          |

| Data | k  | m  | NI/LL | GGMMs ($|a| ≥ 0$) | PFGMMs ($|a| ≥ 0$) | DFGMMs ($|a| ≥ 0$) |
|------|----|----|-------|----------------|-----------------|----------------|
| ‘6’  | 5  | 3  | LL    | -13958        | -13983          | -14903         |
|      |    |    | NI    | 578           | 278             | 288            |
|      |    |    | Time(s) | 110.50        | 55.89           | 55.343         |
| ‘8’  | 6  | 4  | LL    | -7393.7       | -7405.5         | -7755.7        |
|      |    |    | NI    | 666           | 211             | 224            |
|      |    |    | Time(s) | 102.81        | 36              | 35.89          |
| ‘a’  | 7  | 3  | LL    | -8837.9       | -8852.7         | -9470.3        |
|      |    |    | NI    | 770           | 310             | 330            |
|      |    |    | Time(s) | 150.3         | 63.98           | 64.98          |
| ‘B’  | 7  | 4  | LL    | -8753.4       | -8761.8         | -9164.2        |
|      |    |    | NI    | 209           | 173             | 171            |
|      |    |    | Time(s) | 41.84         | 36.05           | 35.45          |
| ‘R’  | 7  | 3  | LL    | -8974.2       | -8982.1         | -9601.9        |
|      |    |    | NI    | 649           | 257             | 197            |
|      |    |    | Time(s) | 133.187       | 55              | 40.57          |

All AP 100% 53.33% 43.11% All AP 100% 48.48% 47.70%
Time 100% 69.29% 47.63% Time 100% 51.12% 48.71%
Table 4.3 shows more examples including the datasets ‘6’, ‘8’, ’a’, and ’B’, whose datasets contain curve manifolds among the numbers and characters. The number of components is chosen according to the dataset’s distribution and facts such as ensuring datasets are fitted well and components are not too many. Degree of fuzziness is decided in accordance with (Pei et al., 2001) illustrating that optimal value of $m$ locates around the range of $[1.5 - 3.5]$. The results of log-likelihoods do not change much when comparing the FGMMs, and the big changes of the number of iterations and also computational time indicate that the fuzzy algorithms are almost double as efficient as conventional GMMs and generalized GMMs. The values of ‘time’ present that probability based FGMMs and distance based FGMMs use 69.29% and 47.63% computational cost of conventional GMMs, and 51.12% and 48.71% computational time of the generalized GMMs.

The above examples confirm that the proposed fuzzy algorithms can achieve similar level of likelihoods as conventional GMMs and generalized GMMs with double efficiency in terms of convergence. Comparing probability based FGMMs with distance based FGMM, though they both have abilities to fit the datasets with acceptable likelihoods, statistical data in Table 4.3 indicates that distance based FGMMs outperform the other in terms of computational cost.

In this chapter, one of the objectives is to save the computational time of GMMs, so the ‘fast convergence’ is said to be compared with the convergence speed of GMMs (including normal GMMs and generalized GMMs) rather than K-means whose objective function is a linear function (‘m=1’). In GMMs (including normal GMMs and generalized GMMs in this chapter), the ‘$w_{it}$’ is the posteriori probability like the membership ‘$u_{it}$’ in FCM. After introducing a weighting component on the membership, in FGMMs, the relationship between ‘$w_{it}$’ and ‘$u_{it}$’ can be seen from above equations 4.44-4.46. When ‘$m$’($m > 1$) becomes larger, ‘$\frac{m}{m-1}$’ becomes smaller; ‘$m \rightarrow \infty$’, then ‘$\frac{m}{m-1} \rightarrow 1$’; at that time FGMM turns into GMM. Therefore, the smaller ‘$m$’($m > 1$)’ is, the larger ‘$u_{it}$’ changes, and the fuzzier FGMM becomes. However when the ‘$m$’ is too small, the fitness would become too low. Therefore, the balance between fitness (performance) and fuzziness (convergence speed) should be concerned. The experiments showed that proper ‘$m$’ can achieve a much faster convergence speed with a satisfied fitting degree.
4.5 Fuzzy Gaussian Mixture Models based Recognition

Chapter 4.3 has presented the proposed FGMMs including PFGMMs and DFGMMs by refining the dissimilarity function and maintaining the exponential relationship between membership and distance. They demonstrated their performance in modeling different characters, but how to apply them in learning the human hand motions is still not addressed. In this section, the algorithm which is used to learn and recognize human hand motions is proposed.

Figure 4.12: An example of DFGMMs based in-hand manipulation motion learning

The captured data of human hand motions are discrete time series. Figure 4.12 shows an example of the angle trajectories of one in-hand manipulation. The motion was repeated ten times, thus ten such angle trajectories were recorded. Since the experiments in section 4.4 showed that DFGMM outperforms PFGMM in terms of the convergence speed, DFGMM is chosen to learn human hand motions in the following chapters and experiments.

To learn these trajectories, five components are selected and the proposed EM algorithm 1 is employed. The learning result is shown as the blue bent ellipses in figure 4.12. Their centers and covariances are indicated by the intersection and the lengths of the two-axes in the components. The ten angle trajectories, which consist of 2,000 points in this example, are thus statistically represented by the five components with five centers and five covariances.
4.5 Fuzzy Gaussian Mixture Models based Recognition

For recognizing the testing motions, the algorithm of equation 4.48 and equation 4.49 are constructed here. Supposing there are $a$ attributes in the human hand motion dataset. The dissimilarity of the testing motion and the trained model of DFGMM is defined by the normalized attribute distance between the components and the testing motion as:

$$Dis = \frac{1}{a} \sum_{k=1}^{a} \text{AvgDis}^{(k)}$$  \hspace{1cm} (4.48)

where $k$ is the number of the attributes; $\text{AvgDis}^{(k)}$ is the average distance of the $k^{th}$ attribute and is calculated by

$$\text{AvgDis}^{(k)} = \frac{1}{Cnum^{(k)}} \sum_{i=1}^{Cnum^{(k)}} |(T\text{motion}^{(k)}(t_i) - \text{Center}^{(k)}(i))|$$  \hspace{1cm} (4.49)

where $Cnum$ is the number of the Gaussian components which is set before applying EM algorithm; $i = 1, ..., Cnum$ is the index of the Gaussian components; $t_i$ is the time point of $i^{th}$ Gaussian component; $T\text{motion}(t_i)$ is the angle value or finger tip position of the testing hand motion at $t_i$; $\text{Center}(i)$ is the $i^{th}$ model center. The corresponding motion points at the time $t_i$ are identified according to the time points of the model centers. The corresponding points are in the same time as these DFGMM’s centers along the dash line. If there is not the same time points as centers, the nearest time points to the dash line will be regarded as the corresponding points. Then the average distances

![Figure 4.13: An example of DFGMMs based in-hand manipulation motion recognition](image)

motion points at the time $t_i$ are identified according to the time points of the model centers. The corresponding points are in the same time as these DFGMM’s centers along the dash line. If there is not the same time points as centers, the nearest time points to the dash line will be regarded as the corresponding points. Then the average distances
between the centers and these points are calculated, as the black lines shown in the Fig 4.13. It calculates the average distance between testing motions and DFGMM centers. The testing sample is recognized as the corresponding grasp class with the minimum dissimilarity.

DFGMM has been applied to recognize human hand motions including grasps and in-hand manipulations under different conditions. The experimental results will be presented and compared with those of TC, GMM, FEC and FEC in Chapter 6.

4.6 Summary

In this chapter, generalized GMMs with an EM algorithm first have been proposed by integrating conventional GMMs and active axis curve GMMs for fitting non-linear datasets. Then, based on the generalized GMMs, FGMMs have been proposed including probability based FGMMs and distance based FGMMs by refining the dissimilarity functions. Probability based FGMMs adapt the dissimilarity function of multiplicative inverse of probability density function, while distance based FGMMs refine the dissimilarity function which limits the degree of fuzziness only on the exponential distance. Experiments have been conducted on conventional Gaussian based datasets, generalized Gaussian based datasets and datasets of written characters. Simulation results have shown that not only do the proposed fuzzy algorithms have the capabilities of fitting data as conventional GMMs or generalized GMMs but they also can reduce to about half of their computational time. Comparing probability based FGMMs with distance based FGMM, the latter outperforms the former in terms of computational efficiency. Due to the nature of the fast computation and non-linear fitting of the proposed FGMMs, they would break new ground for extending GMMs’ application area to the fields such as data mining with huge datasets and real-time systems. Finally, the recognition algorithm based on DFGMM for human hand motions is proposed and its experiments will be shown in chapter 6.
Chapter 5

Fuzzy Empirical Copula

5.1 Introduction

The information in the world is becoming more and more electronic. Due to the improvements in data collection and storage during the past decades, huge amounts of data can lead to the problem of information overload (Edmunds & Morris, 2000) for many researchers in domains such as engineering, economics and astronomy. The increase of the number of dimensions associated with each observation and growth of the sampling time points are the main reasons of information overload. In many cases, datasets contain not only useful messages but also considerable trivial and redundant information both in the dimensions (attributes) and samples. How to remove the redundant information and maintain the important information is crucial in many applications. Two important methods are normally employed to solve this problem: dimensionality reduction and clustering. The formal reduces trivial attributes, maintaining the number of samples, while the latter eliminates the redundant samples without changing the number of attributes.

There are various traditional and current state of the art dimensionality reduction methods to solve the above problem. Principal Component Analysis (PCA) was invented in 1901 by Pearson (1901) and is mostly used for dimensionality reduction in a dataset by retaining the characteristics of the dataset that contribute most to its variance. It keeps lower-order principal components and ignores higher-order ones. Such lower-order components often contain the “most important” aspects. Like PCA,
5.1 Introduction

Factor analysis (FA) is another second-order method (Mardia et al., 1979). FA becomes essentially equivalent to PCA if the “errors” in the FA model are all assumed to have the same variance. These second-order methods require classical matrix manipulations and assumption that datasets are realizations from Gaussian distributions. For non-Gaussian datasets, higher-order dimension reduction methods such as Projection Pursuit (PP) (Friedman et al., 1988) and Independent Component Analysis (ICA) (Comon et al., 1994) are introduced. Additionally, non-linear PCA can also deal with non-Gaussian datasets using non-linear objective functions to determine the optimal weights in principal (Karhunen et al., 1998). Its resulting components are still linear combinations of the original variables, so it can be regarded as a special case of ICA. Other non-linear methods such as Principal Curves (PC) (Hastie & Stuetzle, 1989) and Self Organizing Maps (SOM) (Kohonen, 2001) can be thought to be non-linear ICA (Karhunen, 2001) in that they replace the linear transformation of ICA with a real-valued non-linear vector function. Curvilinear Component Analysis (CCA) is a relatively new non-linear mapping method, being improved from Sammon’s mapping by Demartines & Herault (1997). It uses a new cost function able to unfold strongly non-linear or even closed structures, which significantly speeds up the calculation and interactively helps users control the minimized function. However, more parameters should be considered for most of these high-order and non-linear dimensionality reduction methods and their performances strongly depends on complex adjustments of these parameters, for instance there are three parameters in CCA: the projection space dimension and the two time decreasing parameters.

However, dimensionality reduction methods can not be used during the estimating of a data dependence structure, because the dependence structure includes all the interrelations of the attributes and high-order attributes are not supposed to be ignored.

Clustering is the classification of objects into clusters so that objects from the same cluster are more similar to each other than objects from different clusters. It can effectively reduce the data samples, so it is suitable for reducing the redundant information when estimating data dependence structure. The most common algorithms include K-means (Hartigan & Wong, 1979a), fuzzy C-means (Dunn, 1973), and fuzzy C-means-derived clustering approaches such as fuzzy J-means (Belacel et al., 2002) and fuzzy SOM (Vuorimaa, 1994), which construct clusters on the basis of the pairwise distance between objects, so that they are incapable of capturing non-linear relationships.
and thereby fail to represent a dataset with non-linear structure. Hierarchical clustering is another important approach but suffers from lack of robustness, non-uniqueness, and inversion problems (Tamayo et al., 1999). GMM is based on the assumption that datasets are generated by a mixture of Gaussian distributions with certain probability. But this assumption is not always satisfied for general datasets even after various transformations aimed at improving the normality of the data distribution (Fu & Medico, 2007; Yeung et al., 2001).

Copula is a general way of formulating a multivariate distribution with uniform marginal distributions in such a way that various general types of dependence can be presented (Fermanian & Scaillet, 2003). The copula of a multivariate distribution can be considered as describing its dependence structure as opposed to the behaviour of each of its margins. It is a good way of studying scale-free measures of dependences among variables and also a good starting point for constructing families of bivariate distributions (Nelsen, 2006). Sklar’s theorem (Sklar, 1959) states that a multivariate distribution function can be represented by a copula function which binds its univariate margins. Further, empirical copulas were introduced and first studied by Deheuvels (1979, 1981), which can be used to study the interrelations of marginal variables with unknown underlying distributions.

The copula approach has many advantages (Kolev et al., 2006) and has been used widely in finance (Dias & Embrechts, 2004; Embrechts et al., 2003) and econometrics (Hu, 2006; Trivedi & Zimmer, 2006). Kołosarova et al. (2006) defined a new copula called discrete copulas on a grid of the unit square and showed that each discrete copula is naturally associated with a bistochastic matrix. De Baets & De Meyer (2007) also presented a general framework for constructing copulas, which extended the diagonal construction to the orthogonal grid construction.

Simultaneously, empirical copula (EC) has gained an increasing amount of attention recently. Dempster et al. (2007) constructed an EC for Collateralized debt obligation tranche pricing and achieved a better performance than the dominant base correlation approach in pricing non-standard tranches. Ma & Sun (2008) proposed a Chow-Liu like method based on a dependence measure via empirical copulas to estimate maximum spanning product copula with only bivariate dependence relations, while Morettin et al. (To appear) proposed wavelet estimators based on empirical copulas which can be used for independent, identically distributed time series data.
5.2 Dependence Structure Estimation via Empirical Copula

It is evident, however, that the efficiency of EC is outstandingly poor though it provides effective performance on data dependence structure estimation. It is common that natural datasets are represented by tremendous storage size, and it is impossible to process them using EC in most cases. In order to overcome this problem, an algorithm named fuzzy empirical copula (FEC) is proposed which integrates fuzzy clustering with EC. Fuzzy Clustering by Local Approximation of Memberships (FLAME) (Fu & Medico, 2007) is firstly extended into multi-dimensional space, then the FLAME+ is utilized to reduce the sampling data and maintaining the interrelations at the same time before data dependence structure estimation takes over. Flame+ has the ability to capture non-linear relationships and non-globular clusters and can automatically define the number of clusters and identify cluster outliers. Then FEC is applied to recognize human hand motion based on a template matching mechanism. The remainder of the Chapter is organized as follows. Section 5.2 presents copula theory with a focus on dependence structure estimation using EC. Section 5.3 revisits the fundamental of fuzzy clustering by local approximation of memberships; Section 5.4 proposes the FEC algorithm; Two case studies are given in Section 5.5 to demonstrate the effectiveness of the proposed FEC; Section 5.6 proposes the recognition algorithm using FEC including the one-to-one correlation modeling and template matching method. Finally, concluding remarks are presented.

5.2 Dependence Structure Estimation via Empirical Copula

As a general way of formulating a multivariate distribution, copula can be used to study various general types of dependence between variables. Other ways of formulating multivariate distributions include conceptually-based approaches in which the real-world meaning of the variables is used to imply what types of relationships might occur. In contrast, the approach via copulas might be considered as being more raw, but it does allow much more general types of dependencies to be included than would usually be invoked by a conceptual approach. Nelsen (2006) has proven that these measures, such as Kendall’s tau, Spearman’s rho and Gini’s gamma, can be re-expressed
only in terms of copula. Though their direct calculation may have much less computational cost than when using copulas, copula summarizes all the dependence relations and provides a natural way to study and measure dependence between variables in statistics. It is a very important approach since copula properties are invariant under strictly increasing transformations of the underlying random variables. Spearman’s rho and Gini’s gamma are considered in this Chapter. In this section, we firstly revisit the theoretical foundation of copula and EC, then introduce the theorem of calculating Spearman’s rho and Gini’s gamma using bivariate EC, finally analyse the time complexity of the computation.

5.2.1 Copula

A n-dimensional copula is defined as a multivariate joint distribution on the n-dimensional unit cube \([0, 1]^n\) such that every marginal distribution is uniform on the interval \([0, 1]\).

Definition 5.2.1.1. A n-dimensional copula is a function \(C\) from \(I^n\) to \(I\) with the following properties (Nelsen, 2006):

1. \(C\) is grounded, i.e., for every \(u\) in \(I^n\), \(C(u) = 0\) if at least one coordinate \(u_j = 0\), \(j = 1, \ldots, n\).

2. If all coordinates of \(u\) are 1 except for some \(u_j\), \(j = 1, \ldots, n\), then \(C(u) = C(1, \ldots, 1, u_j, 1, \ldots, 1) = u_j\).

3. \(C\) n-increasing, i.e., for each hyperrectangle
\[B = \times_{i=1}^n [x_i, y_i] \subseteq [0, 1]^n\]
\[
V_c(B) = \sum_{z \in \times_{i=1}^n \{x_i, y_i\}} (-1)^{N(z)} C(z) \geq 0 \quad (5.1)
\]

where the \(N(z) = \text{card}\{k \mid z_k = x_k\}\). \(V_c(B)\) is the so called C-volume of \(B\).

Sklar’s Theorem (Sklar, 1959) is central to the theory of copula and underlies most applications of the copula. It elucidates the role that copula plays in the relationship between multivariate distribution functions and their univariate margins.
5.2 Dependence Structure Estimation via Empirical Copula

Sklar’s Theorem 5.2.1.1. Let $H$ be a joint distribution function with margins $F_i(i = 1, 2, \cdots, n)$. Then there exists a copula $C$ such that for all $x_i$ in $\hat{R}$,

$$H(x_1, \cdots, x_n) = C(F_1(x_1), \cdots, F_n(x_n)) \quad (5.2)$$

where $C$ is a $n$-dimensional copula, $F_i$ are marginal distribution function of $x_i$.

If $F_i(i = 1, \cdots, n)$ are continuous, $C$ is unique. If $C$ is a $n$-dimensional copula and $F_i(i = 1, \cdots, n)$ are distribution functions, then the function $H$ defined by equation 5.2 is a joint distribution function with margins $F_i(i = 1, \cdots, n)$. More details can be seen in (Kolev et al., 2006; Nelsen, 2006).

5.2.2 Empirical Copula and Dependence Estimation

The EC is a characterization of the dependence function between variables based on observational data using order statistics theory and it can reproduce any pattern found in the observed data. If the marginal distributions are normalized, the EC is the empirical distribution function for the joint distribution. Priority has been given to bivariate EC due to computational cost. The reason is twofold: one is that the interrelation between every two attributes is the basic relationship in most attributes, and it is practical to use bivariate EC to construct the whole structure of every two attributes’ dependence; the second is that the dependence structure of dataset $X$ including $r$ attributes would have $\binom{r}{2} = \frac{1}{2}r(r-1)$ bivariate interrelations. Bivariate EC is given as follows.

Definition 5.2.2.1. Let $\{(x_k, y_k)\}_{k=1}^n$ denote a sample of size $n$ from a continuous bivariate distribution. The EC is the function $C_n$ given by

$$C_n\left(\frac{k}{n}, \frac{j}{n}\right) = \frac{\text{card}\{(x,y): x_n \leq x(i), y_n \leq y(j)\}}{n} \quad (5.3)$$

where $x(i)$ and $y(j)$, $1 \leq i, j \leq n$, denote order statistics from the sample (Nelsen, 2006).

The EC frequency $c_n$ is given by

$$c_n\left(\frac{k}{n}, \frac{j}{n}\right) = \begin{cases} \frac{1}{n}, & \text{if } (x(i), y(j)) \text{ is an element of the sample} \\ 0, & \text{otherwise} \end{cases} \quad (5.4)$$
5.3 Fuzzy clustering by Local Approximation of Memberships

Note that $C_n$ and $c_n$ are related via

$$C_n\left( \frac{i}{n}, \frac{j}{n} \right) = \sum_{p=1}^{i} \sum_{q=1}^{j} c_n\left( \frac{p}{n}, \frac{q}{n} \right)$$  \hspace{1cm} (5.5)

**Theorem 5.2.2.1.** Let $C_n$ and $c_n$ denote, respectively, the EC and the EC frequency function for the sample $\{(x_k, y_k)\}_{k=1}^{n}$. If $\rho$ and $\gamma$ denote, respectively, the sample versions of Spearman’s rho, and Gini’s gamma (Kruskal, 1958; Lehmann, 1966), then

$$\rho = \frac{12}{n^2 - 1} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[ C_n\left( \frac{i}{n} \cdot \frac{j}{n} \right) - \frac{i}{n} \cdot \frac{j}{n} \right]$$  \hspace{1cm} (5.6)

and

$$\gamma = \frac{2n}{[n^2/2]} \left\{ \sum_{i=1}^{n-1} C_n\left( \frac{i}{n}, 1 - \frac{i}{n} \right) - \sum_{i=1}^{n} \left[ \frac{i}{n} - C_n\left( \frac{i}{n}, \frac{i}{n} \right) \right] \right\}$$  \hspace{1cm} (5.7)

Spearman’s rho and Gini’s gamma are two ways of measuring two variables’ association (Nelsen, 2006). According to the definition and theorem, we can estimate correlations between variables using EC and Spearman’s rho & Gini’s gamma. Suppose the number of objects is $n$ and number of attributes is $r$. For $r << n$, according to the equations 5.3, 5.6 and 5.7, the time complexity of Spearman’s rho or Gini’s gamma is $O(n^3)$.

5.3 Fuzzy clustering by Local Approximation of Memberships

In this section Fuzzy clustering by Local Approximation of Memberships (FLAME) is extended first in terms of dimension and distance functions, then is integrated into EC to enhance its computational efficiency. FLAME was proposed to cluster DNA microarray data (Fu & Medico, 2007). It defines clusters in the relatively dense regions of a dataset and performs cluster assignment solely based on the neighbourhood relationships among objects. One of the FLAME algorithm features is that the memberships of neighbouring objects in the fuzzy membership space are set according to the neighbourhood relationships among neighbouring objects in the feature space. In this chapter, FLAME is extended in terms of dimension and distance function (i.e., FLAME$^+$), which still consists of three main steps of FLAME algorithm: initialization, approximation and assignment.
5.3 Fuzzy clustering by Local Approximation of Memberships

5.3.1 Initialization

The first step, initialization, is to classify three types of objects: Cluster Supporting Object (CSO), cluster outliers and the rest which are named Normal Points (NPs).

Let \( X \) be a \( r \)-dimensional dataset with \( n \) objects. The \( r \)-dimensional distance between two instances is

\[
d_p(x, y) = \left( \sum_{i=1}^{r} |x_i - y_i|^p \right)^{1/p}
\]

(5.8)

where \( x, y \in X; 1 \leq p \leq \infty; d_1 \) is the Manhattan distance, \( d_2 \) is the familiar Euclidean distance, and \( d_\infty \) corresponds to the maximum distance in any dimension. Then the similarity of these two objects is calculated as:

\[
s_{xy} = \frac{1}{d_p(x, y)}
\]

(5.9)

Similarity is the degree of resemblance between two or more objects. There are different ways to calculate the similarity. Since “the density of each object is calculated as one over the average distance to the k-nearest neighbors” in the FLAME clustering algorithm (Fu & Medico, 2007), to make the relation between similarity and density more direct and simple, Eq. 5.9 is chosen here to calculate the similarity.

The K-Nearest Neighbours (KNNs) for each object are defined as the \( k \) objects \((k \leq n)\) with the \( k \) highest similarity. The density of object \( x \) with KNNs can be obtained

\[
Den_p(x) = \frac{k}{\sum_{y \in knn(x)} d_p(x, y)}
\]

(5.10)

where \( knn(x) \) stands for the set of KNNs of the object \( x \).

Subsequently, the set of CSOs is defined as the set of objects with local maximum density, i.e., with a density higher than that of every object in their KNNs. The higher \( k \) is, the less CSOs will be identified, then less clusters will be generated. A density threshold needs defining to find possible cluster outliers, so objects with densities below the threshold are defined as possible outliers.

Each object \( x \) is associated with a membership vector \( p(x) \), in which each element \( p_i(x) \) indicates the membership degree of \( x \) in cluster \( i \)

\[
p(x) = (p_1(x), \ldots, p_m(x)),
\]

(5.11)
5.3 Fuzzy clustering by Local Approximation of Memberships

where \( 0 \leq p_i(x) \leq 1 \); \( \sum_{i=1}^{m} p_i(x) = 1 \); \( m \) is the total number of CSOs and the outlier cluster, \( i.e., m = c + 1 \) where \( c \) is the number of CSOs; Each element of membership vector takes a value between 0 and 1, indicating how much percentage an object belonging to a cluster, or being an outlier.

Based on the density estimation, each CSO is assigned with fixed and full membership to itself to represent one cluster, for example \( p(x) = (0, 1, \ldots, 0) \) indicates that object \( x \) is the second CSO. Each outlier is assigned with fixed and full membership to the outlier group, \( p(x) = (0, \cdots, 0, 1) \), and the NP is assigned with equal memberships to all clusters and the outlier group, \( p(x) = (1/m, \cdots, 1/m) \).

5.3.2 Approximation

The second step is named local/neighbourhood approximation of fuzzy memberships, in which each NP’s fuzzy membership is updated by a linear combination of the fuzzy memberships of its KNNs, while CSOs and outliers maintain the fixed and full memberships to themselves respectively.

The weights defining how much each neighbour will contribute to approximation of the fuzzy membership of that neighbour are estimated in equation 5.12, based on the fact that the neighbours that have higher similarities must have higher weights.

\[
w_{xy} = \frac{s_{xy}}{\sum_{z \in \text{knn}(x)} s_{xz}} \quad (5.12)
\]

where \( y \in \text{knn}(x) \). The membership vector of each NP is approximated according to equation 5.13, minimizing the overall difference between membership vectors and their approximations.

\[
p^{t+1}(x) = \sum_{y \in \text{knn}(x)} w_{xy} p^t(y) \quad (5.13)
\]

The overall local/neighbourhood approximation error is calculated by:

\[
E(\{p\}) = \sum_{x \in X} \left( p(x) - \sum_{y \in \text{knn}(x)} w_{xy} p(y) \right)^2 \quad (5.14)
\]

The iteration of equation 5.13 breaks under the condition that \( E(\{p\}) \) is less than a predetermined threshold.
5.3 Fuzzy clustering by Local Approximation of Memberships

5.3.3 Assignment

The final stage is to assign each object to the cluster based on its fuzzy membership. Usually, one cluster contains the objects that have higher membership degrees in this cluster than other clusters.

An example of FLAME$^+$ is provided in Fig. 5.1, where a dataset with 600 objects is randomly generated from a 3 dimensional distribution. FLAME$^+$ is applied to this dataset and three groups of objects are clustered as outliers, CSOs and NPs.

![Colormap of the clusters](image)

Figure 5.1: Clustering random 3D Euclidean positions using FLAME$^+$. The star points in black are the centres of the clusters (CSO); points labeled with triangles are the outliers; the colour range of NPs represents their membership degrees.

Accurately calculating the entire time complexity of FLAME$^+$ is very challenging in that each iteration of local/neighbourhood approximation depends on the error threshold. However, it is necessary to analyse the complexity of the first step of the algorithm. Suppose the number of objects is $n$, number of attributes is $r$, its CSOs’ number is $c$ and number of nearest neighbours is $k$. For $r << n$ and $k << n$, the time
complexity of the initialization is $O(n^2)$. An empirical study of the time complexity of FLAME$^+$ compared with other algorithms is performed to illustrate that FLAME$^+$ has significant computational advantage over hierarchical clustering, fuzzy C-means and fuzzy SOM, an exception is K-means (Fu & Medico, 2007). Flame$^+$ is capable of capturing non-linear relationships and non-globular clusters and automatically defining the number of clusters and identify cluster outliers; In addition, compared with K-means and fuzzy C-means, the centres of FLAME$^+$ are real instances in the original dataset instead of the centroids of clusters with different traits which probably result in wrong dependence measures. Finally, FLAME$^+$ is also capable of dealing with a free-distributed dataset, which is not always true for algorithms like Gussian Mixture Models. In Section 5.5, FLAME$^+$ and K-means are compared in the context of FEC.

5.4 Fuzzy Empirical Copula

The aim here is to develop an algorithm which can efficiently reduce the computational cost of EC by filtering out redundant information in the sample. In addition, this algorithm should also be capable of dealing with arbitrary-distributed datasets in order to inherit the main advantage of EC for data structure estimation. FLAME algorithm is selected for this purpose in that it not only fulfills the above requirements but also possesses the merit of few parameters, i.e., the number of nearest neighbours and the value of the outlier’s threshold.

It is evident that samples with higher densities are more reliable when used to represent whole samples in such a way that the main feature of the whole sample is maintained. The FLAME algorithms have the capability of identifying those “special” sampling points based on the objects’ density analysis. The “special” points are represented by the CSOs with the highest densities in all clusters. Therefore, the FEC algorithm is proposed here for achieving the following: high dimension FLAME algorithm is employed to identify characteristic feature points, and dependence structure, which is then estimated via EC.

Let $X$ be $r$-dimensional dataset with $n$ objects:

$$X = \begin{pmatrix}
x_{11} & \cdots & x_{1n} \\
\vdots & \ddots & \vdots \\
x_{r1} & \cdots & x_{rn}
\end{pmatrix}$$
so the $i_{th}$ object is represented by the $i_{th}$ column in matrix $X$: $x_i = [x_{1i}, x_{2i}, \cdots, x_{ri}]^T$ and the $j_{th}$ attribute of $X$ is defined as the $j_{th}$ row: $x(j) = [x_{j1}, x_{j2}, \cdots, x_{jn}]$. The dependence structure here is defined as the whole structure of every two attributes’ dependence which can be calculated by bivariate EC, because the interrelation between every two attributes is the most basic relationship in several attributes. Given interrelations between every two attributes, relations of three or more attributes would be derived from their dependence structure which would have $\binom{r}{2} = \frac{1}{2}r(r-1)$ interrelations ($r$ is the number of attributes).

In FEC, firstly FLAME$^+$ reduces the samples from $n$ objects to $c$ CSOs, and then EC analyses the dependence of every two attributes in the derived CSO matrix. The first step can be considered as the operation on the column and the latter on the row. For ideal performance of the proposed algorithm, one of outputs of FLAME$^+$, CSOs, is computed leading to efficient computation. That is to say we only have to implement the first step in the FLAME algorithm which has less time complexity than EC and only one parameter, the number of neighbours, is required since the threshold works only for outliers. The Spearman’s rho and Gini’s gamma of the CSOs would be

$$\rho(u, v) = \frac{12}{c^2-1} \sum_{i=1}^{c} \sum_{j=1}^{c} \left[ C_{c}^{(uv)} \left( \frac{i}{c}, \frac{j}{c} \right) - \frac{i}{c} \cdot \frac{j}{c} \right]$$

and

$$\gamma(u, v) = \frac{2n}{[n^2/2]} \left\{ \sum_{i=1}^{c} C_{c}^{(uw)} \left( \frac{i}{c}, 1 - \frac{i}{c} \right) - \sum_{i=1}^{c} \left[ \frac{i}{c} - C_{c}^{(uv)} \left( \frac{i}{c}, \frac{i}{c} \right) \right] \right\}$$

where $u \in [1, \cdots, r]$ and $v \in (u, \cdots, r)$; $C_{c}^{(uw)}$ is the bivariate EC of the $u_{th}$ and $v_{th}$ attributes with $c$ objects, and $C_{c}^{(uv)} = C_{c}^{(vu)}$.

The optimization is designed to automatically identify the optimized number of neighbours with acceptable errors. The number of nearest neighbours is increased by one at every step during the optimization until the proper number of neighbours is identified. The optimization stops when the overall error of Spearman’s rho or Gini’s gamma in the equation is under the preset error threshold. The pseudo-code of FEC is proposed in algorithm 2 and its ‘EmpSG’ function is proposed in algorithm form 3.
5.4 Fuzzy Empirical Copula

Algorithm 2 Fuzzy Empirical Copula algorithm

Require: $X = \{x_1, x_2, ..., x_n\}$ \{X is a r dimensional dataset with n objects and $r << n\}

Require: $[\rho_{all}, \gamma_{all}]$ \{the Spearman’s rho and Gini’s gamma of the original data X\}

1: for all $i$ such that $1 \leq i \leq n$ do
2:     for all $j$ such that $1 \leq j \leq n$ do
3:         $d_2(x_i, x_j) \leftarrow$ equation \{calculate the Euclidean distance between two objects using equation 5.8\}
4:         end for
5:     end for
6:     $K = 0$ \{$K$ is the number of nearest neighbours under consideration\}
7:     repeat
8:         $K = K + 1$
9:         for all $i$ such that $1 \leq i \leq n$ do
10:             $Den_2(x_i) \leftarrow$ equation \{get the density of every object\}
11:         end for
12:     c = 0 \{the number of CSOs\}
13:     for all $i$ such that $1 \leq i \leq n$ do
14:         if $Den_2(x_i) \geq \max(Den_2(y_i))$ where $y_i \in knn(x_i)$ then
15:             $c = c + 1$
16:             $CSO_c \leftarrow x_i$ \{get CSOs which have the local maximum densities\}
17:         end if
18:     end for
19:     for all $u$ such that $1 \leq u \leq r$ do
20:         for all $v$ such that $(u + 1) \leq v \leq r$ do
21:             $[\rho(u, v), \gamma(u, v)] \leftarrow EmpSG(CSO(u), CSO(v))$ \{CSO(i) is the $i^{th}$ attribution of CSO, EmpSG is a function to calculate the data’s Spearman’s rho and Gini’s gamma as showed in algorithm 3, $\rho_{r \times r}$ is the matrix of Spearman’s rho and $\gamma_{r \times r}$ is the matrix of Gini’s gamma\}
22:         end for
23:     end for
24:     $error = \| [\rho_{all}, \gamma_{all}] - [\rho_{r \times r}, \gamma_{r \times r}] \|$ \{take the Euclidean distance of the $\rho$ and $\gamma$ as the overall error\}
25:     until $error \geq$ threshold \{threshold is the threshold of overall error decided according to the original dataset\}
Algorithm 3 Function of \textit{EmpSG} for Spearman’s rho and Gini’s gamma

\textbf{Require:} \(\text{CSO}(u), \text{CSO}(v)\) \{two attributes in CSO\}

\textbf{Ensure:} \(SP, GI\) \{Spearman’s rho and Gini’s gamma of above two attributes\}

1: \(x = \text{CSO}(u); y = \text{CSO}(v)\) \{\(x\) and \(y\) are two vectors with \(c\) elements\}

2: \(x' = \text{sort}(x); y' = \text{sort}(y)\) \{\(x'\) and \(y'\) are the order statistics of \(x\) and \(y\)\}

3: \textbf{for all} \(i\) such that \(1 \leq i \leq c\) \textbf{do}

4: \textbf{for all} \(j\) such that \(1 \leq j \leq c\) \textbf{do}

5: \(\text{num} \leftarrow 0\) \{initialization\}

6: \textbf{for all} \(t\) such that \(1 \leq t \leq c\) \textbf{do}

7: \textbf{if} \(x(t) \leq x'(i)\) and \(y(t) \leq y'(j)\) \textbf{then}

8: \(\text{num} \leftarrow \text{num} + 1\)

9: \textbf{end if}

10: \textbf{end for}

11: \(\text{EC}(i, j) \leftarrow \text{num} / c\)

12: \textbf{end for}

13: \textbf{end for}

14: \(SP \leftarrow 0\) \{the return value of Spearman’s rho\}

15: \(GI \leftarrow 0\) \{the return value of Gini’s gamma\}

16: \textbf{for all} \(i\) such that \(1 \leq i \leq c\) \textbf{do}

17: \textbf{for all} \(j\) such that \(1 \leq j \leq c\) \textbf{do}

18: \(SP \leftarrow SP + \text{EC}(i, j) - (j \ast i) / (c \ast c)\)

19: \textbf{end for}

20: \textbf{if} \(i \neq b\) \textbf{then}

21: \(GI \leftarrow GI + \text{EC}(i, c - i) - i / c + \text{EC}(i, i)\)

22: \textbf{end if}

23: \textbf{end for}

24: \(SP \leftarrow SP \ast 12 / (c \ast c - 1)\)

25: \(GI \leftarrow 2 \ast c / (c \ast c / 2) \ast (GI - 1 + \text{EC}(c, c))\)

26: \textbf{return} \(SP\) and \(GI\)
5.5 Validation

Experiments are conducted in this section, and results and discussions are provided for evaluating the effectiveness and efficiency of FEC. After a brief explanation of the datasets, EC and FEC are employed respectively to estimate the dependence structures of the datasets. The section is concluded with the roles that clustering algorithms play in FEC.

5.5.1 Testing Datasets

Figure 5.2: The interrelations of length, diameter, whole weight and shell weight of abalone dataset

Abalone (Nash et al., 1994) dataset from UCI machine learning repository (Asuncion & Newman, 2007) was selected to evaluate the proposed algorithm in this section. The abalone dataset was used to predict the age of abalone from the physical measurements such as weight and length, as it is not a trivial task to get their ages by counting the number of rings in their bodies through a microscope. 4177 abalones are sampled with 9 attributes in this dataset. Fig. 5.2 shows interrelations of length, diameter, whole weight and shell weight. This dataset could be regarded as 9 dimensional data with 4177 samples in which some measurements are intrinsically interrelated. The
dataset contains strong non-linear dependences between attributes. Priority herein is given to the sampling density and the interrelations among attributes. Supposing one dataset contains $s$ attributes, its dependence structure includes $\binom{s}{2}$ interrelations of every two attributes among this dataset, which means the abalone dataset has two dependence structures of 36 interrelations to be analyzed.

5.5.2 Dependence Structure Estimation via Empirical Copula

![Figure 5.3: The result of dependences of 36 correlations of the original abalone dataset](image)

Spearman’s rhos and Gini’s gammas are calculated according to equations 5.6 and 5.7 via EC using the two above datasets. The results of abalone’s dependences of 36 correlations for 9 attributes are listed in the Fig. 5.3. The whole computation time for abalone is 27226 seconds, which is unrealistic for related applications.

5.5.3 Dependence Structure Estimation via Fuzzy Empirical Copula

The proposed FEC is employed in this section to reduce computation time for dependence structure analysis of these two datasets. However, when more nearest neighbours are considered, the fewer CSOs will appear. Thus the calculation of FEC will be more efficient but at the cost of lower accuracy. The proposed FEC has the ability to identify...
the proper number of nearest neighbours, which guarantees the fast computation with the overall error under the preset threshold. The threshold for Spearman’s rho from equation 5.15 is predefined as,

\[ \text{threshold} = p \times \binom{s}{2} = \frac{ps(s-1)}{2} \]  

(5.17)

where \( \binom{s}{2} \) is the number of interrelations, combining 2 attributes out of \( s \) attributes; \( p \) is the average error percentage for each interrelation. Different threshold results in different computational time, and \( p \) is defined to take a value in the range from 0.5% to 1% according to the different features of datasets. From the above results of the two datasets using EC, \( p \) is predefined as 0.6% for abalone dataset, which indicates that abalone dataset has the thresholds of 0.228.

Figure 5.4: Result of sampled abalone data using 12-neighbour-FLAME$^+$

Under the overall error threshold of 0.228 for Spearman’s rho, FEC with 12 nearest neighbours has the lowest computation time. Thanks to the FLAME$^+$'s density sampling, when the number of nearest neighbours is 12, the number of data instances is reduced to 100 from 4177 (Fig. 5.4). The 36 interrelations of the 9 attributes of these 100 CSOs are estimated as shown in red line in Fig. 5.5, where the blue lines are results generated from EC. It illustrates that FEC with 12 nearest neighbours does
5.5 Validation

Figure 5.5: Comparison of Spearman’s rhou and Gini’s gammas. Blue lines are the correlations of EC algorithm while red lines are of FEC algorithm.

not cause unacceptable error to Spearman’s rho and Gini’s gamma compared to EC. However, the computation time of fuzzy EC algorithm is 68 seconds, which is only 0.25 percent of the computational time conducted by EC algorithm.

In order to have a better understanding of the performance of FEC, Fig. 5.6 displays the change of CSO’s number with the growing number of nearest neighbours from 1 to 20. It shows that the numbers of abalone’s CSOs drops exponentially with the growth of the number of nearest neighbours. With the growing nearest neighbours, Fig. 5.7 shows the overall error changes of Spearmans rho and Ginis gamma and Fig. 5.8 presents the time change. The error threshold locates the place between 12 and 13 nearest neighbours. Given a threshold, the error and the computational time can easily be decided from Figs. 5.7 and 5.8.

5.5.4 Comparison of Flame$^+$ to K-means for EC

The reason to choose FLAME$^+$ as the fuzzy clustering algorithm instead of other algorithms (e.g., K-means) in FEC is on the basis of FLAME$^+$’s four main advantages. First it has the ability to capture non-linear relationships and non-globular clusters;
Figure 5.6: The relationship between number of nearest neighbours and number of abalone’s CSOs

Figure 5.7: Change of overall errors of Spearman’s rhos and Gini’s gammas with the growth of number of nearest neighbours in FLAME+ secondly it can automatically define the number of clusters and identify cluster outliers; thirdly, compared with K-means and fuzzy C-means, the centres of FLAME+
5.5 Validation

Figure 5.8: Change of time costed by FEC with the growth of number of nearest neighbours in FLAME⁺

are real instances in the original dataset instead of the centroids of clusters with different traits which probably result in wrong dependence measures. Finally, FLAME⁺ is also capable of dealing with a free-distributed dataset, which is not always true for algorithms like Gaussian Mixture Models.

In order to demonstrate the effectiveness of the proposed FEC, a K-means based EC was constructed which employs the K-means algorithm to cluster the original dataset into a number of subsets and uses centroids as the new dataset. Both of these two methods were applied to the abalone dataset under the same condition, where the cluster number of K-means is same as the abalone’s CSOs number of FLAME⁺, which is listed in the table 5.1.

Table 5.1: Number of clusters corresponding to number of nearest neighbors for K-means

<table>
<thead>
<tr>
<th>Neighbors</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clusters</td>
<td>1574</td>
<td>525</td>
<td>377</td>
<td>287</td>
<td>243</td>
<td>211</td>
<td>185</td>
<td>165</td>
<td>142</td>
<td>128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Neighbors</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clusters</td>
<td>109</td>
<td>100</td>
<td>94</td>
<td>89</td>
<td>88</td>
<td>81</td>
<td>74</td>
<td>69</td>
<td>67</td>
<td>63</td>
</tr>
</tbody>
</table>
Figure 5.9: Change of time cost when comparing the proposed FEC and K-means based Empirical Copula (KEC)

Fig. 5.9 demonstrates the comparison of computational cost by the proposed FEC and K-means based EC, where red curves are the changes of cost time by K-means based EC while blue curves are by the proposed FEC. It presents that both of the two algorithms achieve almost the same performance in saving the computational time.

In Fig. 5.10, with the decreasing number of clusters, the errors caused by K-means based EC fluctuate violently and keep much higher than those by proposed FEC. It illustrates that FLAME\(^+\) outperforms K-means in maintaining the dependence structure though both of them have the almost same performance in reducing the cost time. Thus FLAME\(^+\) is more suitable to be used in FEC. One reason for the above results is that FLAME\(^+\) is capable in dealing with non-linear relationships while K-means is not. Another reason is that FLAME\(^+\) considers the real objects CSOs, which are the samples in the datasets while K-means considers centroids which are virtual objects beyond the datasets. Centroids not belonging to the datasets may have different traits which probably result in wrong dependence measures.
5.6 Recognizing Human Hand Motion via Fuzzy Empirical Copula

In this section, FEC is introduced into the field of recognizing human hand motions for the first time. The motion template consists of one-to-one correlations among all the finger angles, and the matching algorithm is proposed to identify the testing motions.

5.6.1 One-to-one correlation and motion template

Suppose there are \( m \) variables which could be sEMG features of all channels in every motion, \( C_m^2 \) is the total number of the one-to-one correlations. Let \( \rho_{ij} \) be the Spearman’s rho between \( i \)th and \( j \)th variables, and the motion template is defined as the matrix \( \mathbf{P} \) of Spearman’s rhos:

\[
\mathbf{P} = \begin{pmatrix}
\rho_{11} & \cdots & \rho_{1m} \\
\vdots & \ddots & \vdots \\
\rho_{m1} & \cdots & \rho_{mm}
\end{pmatrix}
\]
where $\rho_{ij} = \rho_{ji}$ when $i \neq j$ and $\rho_{ij} = 1$ if $i = j$. Given $s$ observations for one motion, the template is trained by taking the average of all Spearman’s rho matrices.

$$\hat{P} = \frac{\sum_{i=1}^{s} w_i P_i}{\sum_{i=1}^{s} w_i} \quad (5.18)$$

where $w = [w_1, \ldots, w_s]$ is a weight vector used to store the relative difference of each observation in the estimated template, so that more valid observation may carry larger weight than those with more uncertainties, which may be caused by noise, capturing devices, software and the environment. Fig. 5.11 shows an example of the motion template representing the one-to-one correlations among the finger angles when grasping a big ball.

The matrix $P$ effectively aggregates the dependence relations of $m$ variables into just one $m \times m$ matrix, a highly reduced dimensionality of the feature space. For example, the relation of the second variable with about 200 values and the third variable with about 200 values in one hand motion is represented as $\rho_{23}$ with only one real value between $-1$ and $1$. The relation matrix is naturally uniform, in which the matrix is not dependent on differently sampled trials associated with specific speeds. This makes direct comparisons of relation matrices with differently sampled data feasible and computationally efficient.
5.6.2 Motion Recognition

Motion recognition is straightforward with the proposed template. It is achieved by finding the best match between an observed motion template and pre-trained motion templates. The proposed algorithm is applied on an observed motion to generate its motion template $U = \{q_{ij}|i, j = 1, \cdots, m\}$. Its dissimilarity with the pre-trained template is achieved by

$$D_t = \|U - P\|_t = \left( \sum_{i=1}^{m} \sum_{j=1}^{m} |q_{ij} - q_{ij}|^t \right)^{1/t} \quad (5.19)$$

$D_t$ is $t$-norm distance; $t \geq 1$ and is a real number; usually we take $t \in \{1, 2, \infty\}$ that $D_1$ is the taxicab norm, $D_2$ is the Euclidean norm and $D_\infty$ is the maximum norm. The derived $D_t$ norm infers the dissimilarity between the observed motion and the trained motions. The threshold of the template $P$ is defined as

$$th_P = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} |q_{ij}|}{\alpha}; \quad (5.20)$$

where $\alpha \geq 1$, which indicates the threshold is $100/\alpha$ percent of the whole absolute value of the template. Different datasets may have different $\alpha$ values. Here, we set $\alpha = 10$, then the threshold of the template in fig 5.11 would be 8.33. The matching criterion of the motion recognition is that if $D_t \leq th_P$, the observed motion is recognized as belonging to the trained motion.

This chapter suggests how FEC could be used to recognize human hand motions for the first time using the proposed motion template and matching algorithm. In practice, it has been applied to recognize human hand motions including grasps and in-hand manipulation under different conditions. The experimental result will be presented and compared with those of TC, FGMM, HMM and GMM in Chapter 6.

5.7 Summary

FEC has been proposed to alleviate the computational burden of EC. A high-dimensional FLAME+ has been developed to identify the important objects containing the main features of the entire dataset, then EC has been implemented to estimate the dependence structure of the objects. Abalone dataset from UCI machine learning repository
are employed to evaluate the proposed method. The number of nearest neighbours is the tradeoff factor for handling accuracy and efficiency of data processing. With the preselected error threshold, FEC has the capability of automatically identifying the optimized number of neighbours, which could be used to fast analyse similar datasets. Additionally, nearest neighbours at the range of 0-20 have been used to demonstrate the overall error changes of Spearman’s rho and Gini’s gamma, and the change of computational time. The experimental results have shown that FEC can substantially reduce the computation cost while features of the data are maintained with the preselected error threshold. In addition, we compare FLAME$^+$ with K-means to evaluate the clustering role in FEC and the result has illustrated that FLAME$^+$ outperforms K-means in maintaining the dependence structure of the datasets.

Though Copula has been widely applied to finance problems in the past decades, some areas such as intelligent robotics, artificial intelligence and automation, require EC and its variants to be both efficient and effective even in real time scenarios. FEC has succeeded in overcoming the problem of computation cost of dependence structure estimation via EC. In addition, the algorithm to recognize human hand motions using FEC has been proposed for the first time and the experimental result will be presented in Chapter 6.
Chapter 6

Experiments and Evaluation

6.1 Introduction

Chapter 3 to 5 have proposed Time clustering, Fuzzy Gaussian Mixture Models and Fuzzy empirical copula respectively. However, these need to be fairly evaluated and the performances of the proposed algorithms are compared with HMMs and conventional GMMs. In the fair evaluation, standard datasets and protocols are required. In this chapter, a wide range of scenarios are used to evaluate the proposed algorithms: a) training with one sample from a single subject; b) training with one sample from multiple subjects; c) training with multiple samples from a single subject; d) training with multiple samples from multiple subjects. In this study, two standard datasets, 13 grasps and 10 in-hand manipulations, are employed. For each, there are 20 individual samples per subject for 6 subjects. Since Chapter 4 has demonstrated that distance based FGMM generally outperforms probability based FGMM in terms of convergence speed, distance based FGMM is selected to compare other methods and is referred as FGMM in this chapter. This chapter is organized as follows: Section 6.2 give the details of the experiment set-up including data capturing system, subjects, tasks and procedures; Sections 6.3 to 6.6 present the four experiments and their discussions; Section 6.8 summarizes this chapter.
6.2 Experiment Set-up

A fully instrumented right handed data glove, CyberGlove from Immersion Corporation, has been employed to measure the finger joint angles during the hand movement. It can accurately transform hand and finger motions into real-time digital joint angle data by proprietary resistive bend-sensing technology (Sato et al., 2004). The sampling frequency of CyberGlove system in the experiments was set to 100Hz without any filter. The joint angles are measured by 22 bend sensors, which include three flexion sensors per finger, four abduction sensors, a palm-arch sensor, and sensors to measure flexion and abduction of the wrist, as shown in Fig. 6.1. Each sensor is extremely thin and flexible and is virtually undetectable in the lightweight elastic glove. CyberGlove is calibrated through the software from the company, and displayed by VirtualHand Studio software, which converts the data into a graphical hand mirroring the subtle movements of the physical hand.

Figure 6.1: Positions of the recorded hand joint angles
6.2 Experiment Set-up

Six (2 female, 4 male) healthy right-handed subjects volunteered for the study, their ages range from 23 to 35 with the average is 29.5 years. All subjects practiced to manipulate different objects. After initial practicing each test grasp or in-hand manipulation is repeated 20 times for every subject. Both grasps and in-hand manipulation are considered in the study of object manipulation.

6.2.1 Grasp Motions

According to the human hand grasp taxonomy (Iberall, 1997), hand grasps are classified into 6 atomic types, for instance, power grasps, precision grasps, and circular/prismatic grasps, etc. 13 different grasp gestures were selected as shown in Fig. 6.2 to test the algorithm recognition ability. Cup has been selected three times respectively in motions 3, 4 and 5 with different strategies. In motion 3, only three fingers are used to precisely contact the cup top; in motion 4, five fingers are laterally attached to the cup and cup handle is grasped with three fingers in motion 5. Note: the same grasps were used to test TC (Sec. 3.3.2).

6.2.2 In-hand Manipulation

The other dataset is for in-hand manipulation, which is much more complex than simple grasp motion and associated with the most complex human motor skills. It’s the ability to change the position/orientation or adjust an object within one hand (Fig. 6.3).
6.2 Experiment Set-up

Figure 6.3: Four examples of pencil in-hand manipulation

Table 6.1: 10 types of in-hand manipulations

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open a mobile phone and then close it.</td>
</tr>
<tr>
<td>2</td>
<td>Screw to open a small bottle using only thumb, index finger and middle finger.</td>
</tr>
<tr>
<td>3</td>
<td>Pick up a coin and move it from the fingertip to the palm</td>
</tr>
<tr>
<td>4</td>
<td>Remove the pencil from back to front for writing, as shown in figure 6.3: (a) pencil walking</td>
</tr>
<tr>
<td>5</td>
<td>Pick up a pencil and simply rotate to write, as show in figure 6.3:(c) simple rotation</td>
</tr>
<tr>
<td>6</td>
<td>Pick up a pencil and complexly rotate to write, as show in figure 6.3:(d) complex rotation</td>
</tr>
<tr>
<td>7</td>
<td>Screw to open a big bottle using all five fingers</td>
</tr>
<tr>
<td>8</td>
<td>Roll a small cylinder</td>
</tr>
<tr>
<td>9</td>
<td>Pick up a scissor and cut paper</td>
</tr>
<tr>
<td>10</td>
<td>Pencil flips, as shown in figure 6.3:(b) pencil flips</td>
</tr>
</tbody>
</table>

There was no previously effective solution reported to the problem of recognition of in-hand manipulations. Ten types of manipulations were recorded by Cyberglove (table 6.1), using the same procedure as for grasps (Sec. 6.2.1).
6.2.3 Note on Data Analysis

The data analysis has been considered in the experiments. The assumptions for Analysis of Variance (ANOVA) is a) normality of sampling distribution. The data will cluster around a mean or average. More of the scores are concentrated in the middle. b) Each sample is randomly selected and independent. The samples do not rely upon one another. c) The populations are assumed to have equal standard deviations or variances. (Homogeneity). The variances are relatively equal. However, in these experiments, the a) and b) assumptions are not satisfied, so the ANOVA techniques is not suitable to use here.

6.3 Experiment 1- Training with One Sample from a Single Subject, Comparing within the Subject

Table 6.2: Recognition results of FEC, TC, FGMM, GMM and HMM with only one sample from single subject as training data

<table>
<thead>
<tr>
<th>Method</th>
<th>FEC</th>
<th>TC</th>
<th>FGMM</th>
<th>GMM</th>
<th>HMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition rate</td>
<td>89.06%</td>
<td>75.37%</td>
<td>59.65%</td>
<td>52.67%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>FEC</th>
<th>TC</th>
<th>FGMM</th>
<th>GMM</th>
<th>HMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition rate</td>
<td>94.82%</td>
<td>81.80%</td>
<td>65.61%</td>
<td>61.11%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The first experiment is designed to investigate the recognition performances of the proposed algorithms with only one sample from a single subject for training the models. The trained model is used to identify the other 19 samples from the corresponding subject. For example, the grasp model is trained from the first sample (totally 20 samples) of subject 1 when he is grasping a big ball, and then the model is used to test the other 19 grasps of subject 1 only. In this experiment, 13 grasp models and 10 in-hand manipulation models have been trained for each subject. The models for each subject are only used for the subject itself. No cross testing is taken.
6.3 Experiment 1- Training with One Sample from a Single Subject, Comparing within the Subject

Figure 6.4: Recognition results of different motions from different subjects on grasp dataset with one sample from single subject as training data

Figure 6.5: Recognition results of different motions from different subjects on in-hand manipulation dataset with one sample from single subject as training data

The recognition results are summarized in Table 6.2, showing that FEC approach performs better than the other four methods. FEC achieved 89.06% and 94.82% ac-
6.4 Experiment 2- Training with Six Samples from Multiple Subjects, Comparing Across Multiple Subjects

Accuracy for identifying the motions in these two datasets. TC scores the second high recognition rate, followed by FGMM and GMM. Note that FGMM performs better than GMM in each dataset, HMM has not be able to identify any motion correctly with only one set of training samples. Fig. 6.4 and Fig. 6.5 present the detailed recognition results of different motions from all subjects seperately for grasps and in-hand motion respectively. In Fig. 6.4, grasps 3 and 7 have relatively high accuracy for all the methods except HMM, while grasps 4, 8 and 13 are difficult to recognize. Fig. 6.5 demonstrates that grasp 9 can be identified by all the proposed methods while in-hand motions 4, 5, and 6 are not straigtforward to be recognized.

6.4 Experiment 2- Training with Six Samples from Multiple Subjects, Comparing Across Multiple Subjects

Table 6.3: Recognition results of FEC, TC, FGMM, GMM and HMM with six samples from multiple subjects as training data

<table>
<thead>
<tr>
<th>Method</th>
<th>Grasp dataset</th>
<th>Inhand manipulation dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC</td>
<td>81.90%</td>
<td>85.83%</td>
</tr>
<tr>
<td>TC</td>
<td>76.65%</td>
<td>82.19%</td>
</tr>
<tr>
<td>FGMM</td>
<td>74.89%</td>
<td>72.10%</td>
</tr>
<tr>
<td>GMM</td>
<td>70.17%</td>
<td>64.82%</td>
</tr>
<tr>
<td>HMM</td>
<td>41.45%</td>
<td>58.42%</td>
</tr>
</tbody>
</table>

The second experiment is to investigate how effective these methods are when they are trained with six samples from multiple subjects and used to identify the different subjects’ new samples. There are 13 grasp models and 10 in-hand models totally for all subjects. Each model is trained with 6 samples from 6 subjects. Each subject contributes one training sample. The models, combined with multiple subjects’ contributions, are used to recognize other 19 samples from all subjects. Thus, cross testing is applied.

The experiment results are tabulated in Table 6.3. The results show that FGMM is still slightly more effective than GMM. HMM still gets the lowest recognition rate.
6.4 Experiment 2- Training with Six Samples from Multiple Subjects, Comparing Across Multiple Subjects

Figure 6.6: Recognition results of different motions from different subjects on grasp dataset with six samples from multiple subjects as training data

Figure 6.7: Recognition results of different motions from different subjects on in-hand manipulation dataset with six samples from multiple subjects as training data
6.5 Experiment 3- Training with Multiple Samples from a Single Subject, Comparing within the Subject

although it rises from 0 to 41.45% (Grasp) and 58.42% (In-hand). In both experiments 1 and 2, the proposed method, FEC, consistently outperforming the other four methods with the highest accuracy for both the grasp and in-hand manipulation datasets, is much more effective with a few samples as training data than other methods. This is a substantial advantage that can alleviate the difficulty of lacking training datasets in many applications.

Detailed recognition results of different motions are shown in Figs. 6.6 and 6.7 for grasp and in-hand manipulation datasets respectively. In terms of the grasp dataset, motion 3, precisely grasping the top edge of the cup using 3 fingers, maintains the highest accuracy in the 13 grasps. Motions 2, 4, 8, 12, 13 have relative low recognition rates. Fig. 6.7 presents that motion 9 remains a high recognition rate, and most of the other motions are still not easy to classify in the in-hand manipulation dataset.

6.5 Experiment 3- Training with Multiple Samples from a Single Subject, Comparing within the Subject

This experiment determines whether the number of training samples affects the recognition capability and how much it enhances the performance of the methods for a single subject. An empirical study was designed to evaluate sensitivity of the five methods in terms of change of the rate of training samples. There are 13 grasp models and 10 in-hand manipulation models for each single subject, where each subject repeated 20 times for each motion. Each model for a subject is trained with various percentage of the data ranging from 5% to 95% from the same subject. The trained model is used to test the unused data from the same subject.

The experimental results with various percentage of training data are illustrated in Fig. 6.8 and Fig. 6.9 for the two datasets. They show that performance of the five methods are all improved with the increase of the training datasets and their accuracies stabilize with more than 30% training data. The recognition rate of HMM improves from 0% to more than 90% when training data increase from 5 to 30 percentages and it reaches as high as that of FEC with more than one third training data. Compared with GMM, FGMM achieves a better result with all the various sample rates. TC stays in
6.5 Experiment 3- Training with Multiple Samples from a Single Subject, Comparing within the Subject

Figure 6.8: Recognition results with various training percentages from single subject on grasp dataset

Figure 6.9: Recognition results with various training percentages from single subject on in-hand manipulation dataset
6.6 Experiment 4- Training with Multiple Samples from Multiple Subjects, Comparing Across All Subjects

The results demonstrate that the variation of the training samples does substantially affect performance of these methods. The increase of the training data can result in about 90, 30, 30, 20 and 10 percent increases of the recognition rates for HMM, GMM, FGMM, TC, FEC. It presents that HMM is sensitive to the varying training data more than any of the other four methods. FEC consistently maintains high recognition rates with varying training samples, which emphasizes that the proposed FEC is capable of fast extracting the intrinsic features from limited training samples and is insensitive to varying training samples. It also demonstrates that the recognition rates of the grasp dataset are always higher than that of in-hand manipulation dataset.

6.6 Experiment 4- Training with Multiple Samples from Multiple Subjects, Comparing Across All Subjects

An experiment was conducted to evaluate the effects of different number of training samples on the models’ performance for classifying data from multiple subjects. There are 13 grasp models and 10 in-hand manipulation models for all the six subjects. Each model was trained by mixing samples from all six subjects, and each subject contributes the same amount of samples to the training data. All other samples, which are not included in the training data, are used to evaluate the models trained by multiple subjects’ samples. The percentage of the training data ranges from 5% to 95%.

The experimental results are provided in Figs. 6.10 and 6.11 for grasp and in-hand manipulation datasets respectively. HMM achieves an improvement of the recognition rate from less than 50% to 90% in terms of usage of the training samples. Different from the previous experiment results, there are no obvious changes of recognition rates for the other methods with varying training samples. It is confirmed that HMM is the most sensitive method responding to varying training samples. Though HMM can achieve the highest accuracy with more than 40 percent training samples, the proposed three methods, FEC, TC and FGMM, consistently outperform HMM and GMM when the training samples are less than one fifth due to the insensitiveness to the varying percentage of the training data. It should be noted that FEC achieves more than 80%
6.6 Experiment 4- Training with Multiple Samples from Multiple Subjects,
Comparing Across All Subjects

Figure 6.10: Recognition results with various training percentages from multiple subjects on grasp dataset

Figure 6.11: Recognition results with various training percentages from multiple subjects on in-hand manipulation dataset
accuracy with only 5% training data, which is far better than those of the HMM and GMM. The recognition result of TC is slightly lower than FEC, but it has the robustness of handling data sampling variations. FGMM’s performance is consistently better than that GMM’s thanks to the non-linear characteristic, though both recognition rates undulate intensely.

### 6.7 Comparison of the Time Cost

Table 6.4: Time cost by modeling and recognizing of FEC, TC, FGMM, GMM and HMM

<table>
<thead>
<tr>
<th>Method</th>
<th>FEC</th>
<th>TC</th>
<th>FGMM</th>
<th>GMM</th>
<th>HMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling (s)</td>
<td>0.63</td>
<td>$4.3 \times 10^{-3}$</td>
<td>0.45</td>
<td>0.58</td>
<td>0.11</td>
</tr>
<tr>
<td>Recognizing (ms)</td>
<td>$1.6 \times 10^{-2}$</td>
<td>1.6</td>
<td>$2.8 \times 10^{-2}$</td>
<td>$2.7 \times 10^{-2}$</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 6.4 presents the time cost by by modeling and recognizing of FEC, TC, FGMM, GMM and HMM, when training with one sample and recognizing one testing sample. TC has a very fast modeling process comparing with other four algorithms, while FEC and FGMM are relatively fast in recognizing process. When training with more than one sample, the modeling time will be multiplied with the number of training samples and the recognizing time will be the same.

### 6.8 Summary

In this chapter, experiments, covering different subjects and various scenarios, have been conducted to demonstrate the effectiveness of the proposed framework in comparison with HMMs and GMMs under the same conditions. To fairly evaluate the performances of the proposed approaches with standard datasets and protocols, two datasets consisting of 13 different types of grasps and 10 in-hand manipulations have been employed for the algorithm evaluation.

FEC requires low quantity of training samples and achieves a relatively high recognition rate for both single subject and multiple subjects. Low computational cost
is needed for modeling though its recognition step could be costly since dissimi-
larity comparison takes place at every time point of the test sequence; FGMM is a
component-based modeling method having the capability of modeling datasets with
curve manifolds though it does not have as good recognition rate as TC and FEC. Its
performance substantially depends on the number of Gaussian components; FEC rec-
ognizes hand motion via its data dependency structure among the joint angles of hand
fingers, FEC is the least sensitive to the varying training data samples and achieved the
most satisfactory results for both single subject and multiple subjects with few training
dataset.

Experiment results demonstrated that all the three methods in the framework were
insensitive to the varying training samples to certain extent, which means that these
methods have advantages in some practical applications, especially when requiring a
fast training model process or lacking sufficient training datasets. It is evident that
arranging the three algorithms under a framework allow providing a feasible solution
to hand motion recognition for a wide range of hand scenarios.
Chapter 7

Conclusions

This chapter discusses and summarizes the work, experiment results and main contributions of the thesis. Both the strengths and weaknesses of the proposed algorithms are discussed. Future research work are provided, addressing the weaknesses of the approaches and hypothesizing on how to further apply the developed methodologies.

7.1 Overview and Contribution

Human hand motion recognition plays an important role in human-robot skill transfer and attracts more and more research interests. Many approaches have been proposed or applied to understand human hand motions. However, due to the complexity and dexterity of the human hand, most of the existing methods have limitations of only being effective with restricted subjects, constrained motions or sufficient training samples. Though some methods achieved high recognition rate, e.g. neural network and HMM (Moni & Ali, 2009; Symeonidis), the associated high computational and time complexity has limited their utilizations and practical application. In addition, there is no effective solution reported to recognize the complex hand skill such as the in-hand manipulation.

To address the above problems, this thesis proposed a novel fuzzy framework of a set of recognition algorithms: TC, FGMM and FEC, using numerical clustering, Gaussian pattern and data dependency structure respectively to optimally real-time recognize various human hand motions. A variety of conditions were considered in the experiments: a) datasets including 13 grasps and 10 in-hand manipulations; b) single
subject and multiple subjects, c) varying training samples. Performances of the proposed algorithms were fairly evaluated with the comparison of HMM and GMM under the same conditions. Experimental results revealed a number of interesting points and the discussions of these three algorithms, which include the conditions and requirements for using different techniques, are summarized as follows.

The extended TC is capable of generating the desired trajectory automatically since its modeling mechanism is based on the fuzzy modeling of trajectories and its output is the corresponding trajectory point. The recognition algorithm itself can instinctively identify the start point and end point of the motion. It requires low quantity of training samples and achieves a relatively high recognition rate for both single subject and multiple subjects. Low computational cost is needed for modeling. This method is applicable to motion planning directly transferred from the recognition result.

FGMM is a component-based modeling methods with a fast convergence process. With the bend principle axis, its component has the ability of modeling datasets with curve manifolds. The efficiency of convergence has been improved by the introduction of a weighting exponent on the fuzzy membership, which saves computational time of its EM algorithm. Low training samples were required, and its accuracy is always better than that of GMM, though it is lower than those of TC and FEC. Recognition algorithm was fast and the model storage space is small since the number of components are much less than points of the trajectory. Gaussian Mixture Regression (GMR) could be employed for generating desired trajectory.

FEC studies the dependence structure among the finger joint angles of the motion to recognize human hand motions. Fuzzy sampling has been adopted for a fast modeling process. FEC is the least sensitive to the varying training samples among these methods and achieves very satisfactory results for both single subject and multiple subjects with few training dataset. Both modeling and recognizing were fast and model storage space with the dependence structure was small as well. It is most useful for the applications which needs fast modeling and lacks training samples.

The main contributions of this thesis can be summarized as follows:

1. The main contribution is a novel fuzzy framework, which consists of a set of recognition algorithms for human hand motion recognition, which provides different methodologies with different level features. It is capable of fast modeling
various hand motions from different subjects with limited training samples. It provides an effective solution for human hand motion recognition in different practical applications. The extended TC is applicable to motion planning directly transferred from the recognition result. FGMM is applicable to the applications which have a small model storage space and require a method to generate the desired trajectory. FEC can be used in the applications requiring high recognition rate and no desired trajectory with limited training samples.

2. Fuzzy approaches are proposed with improved learning efficiency. The extended TC is a fast fuzzy time-modeling approach with numerical value as output. With the extended degree of membership, TC is capable of modeling both the time instance and different learning motion. The extended TC can model both repeated motions from the same subject and the similar gestures from various subjects; FGMM is proposed based on the generalized GMM with a refined dissimilarity function. The proposed FGMM not only possesses non-linearity but also has a computationally inexpensive convergence process. Two new types of FGMMs, which are probability based FGMMs and distance based FGMMs, have been proposed. FGMMs have better efficiency than conventional GMMs and generalized GMMs, and the distance based FGMMs outperform probability based FGMMs in terms of the learning efficiency; FEC is proposed by integrating FLAME+ with EC and has the ability to save huge computational cost of estimating the dependence structure. FLAME+, which is extended into multi-dimensional space, is utilized to reduce the number of sampling data and maintaining the interrelations at the same time before data dependence structure estimation takes over. FEC is an effective dependence estimation method with a greatly increased efficiency. These improvements enable the proposed framework to optimally real-time or near real-time learn or recognize human hand motions.

3. Experiment results have been presented to compare the different methodologies. Both grasp motions and in-hand motions have been employed to test the proposed framework. Different subjects and varying training samples are also considered. The propose framework is not only capable of identifying grasp motions but also has ability to recognize in-hand manipulations with a better performance than HMM and GMM.
7.2 Future Work

In the following section we discuss the limitations of our work and some directions for the future work.

7.2.1 Further Development of The Proposed Fuzzy Framework

The efficiency and effectiveness of the proposed fuzzy framework has been demonstrated through experiments on recognizing human hand motions. This framework could be improved in the following aspects:

1. TC utilizes motion trajectories as output and is capable of naturally segmenting the motion sequences. However, computational and time cost in its recognition step is relatively high compared to those of FGMM and FEC since the dissimilarity comparison takes place at every time instant of the test sequence. Further strengthening the efficiency of TC in terms of the fast recognition efficiency will be one of the future research directions.

2. The performance of FGMM depends on the number of components. Insufficient components will result in losing important information about the distribution of the trajectories. However when too many components are considered, local components corresponding to small group points would cause noises and computational time would be increased dramatically. A method to automatically decide the number of components for different motions is a topic for future research work. Additionally, it is an important contribution to extend the proposed framework to other types of probabilistic graphical models, such as factor analysis and Ising models (Jordan, 2004).

3. FEC studies the dependence structure of finger angles as the matching template in the recognition algorithm. Nevertheless, it is difficult to directly generate the desired trajectory from the dependence structure. The way to link the desired motion trajectory to the recognition result of FEC or the dependence structure needs further investigation.
7.2 Future Work

7.2.2 Human Hand Recognition in Unconstrained Scenes

Hand motion recognition is a challenging task because hand motion lacks a clear categorical structure: the same human hand motion can be classified into several categories when the motion occurs in different scenes. Furthermore, two different complex motions may contain the same simple hand motions. The more complex the hand motion is, the more difficult the recognition becomes. The topology of human hand motions will provide a feasible way to construct complex hand skills by combining the fundamental hand motions. Research on hand motion patterns constructed by fusing qualitative description is another future research direction for recognizing the hand motion in complex unconstrained scenes (Chan & Liu, 2009; Liu et al., 2008). On the basis of those future work, the proposed fuzzy framework could be applied to applications such as EMG based human motion recognition for prosthetic hands and human-robot skill transfer.
7.2 Future Work
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


140
REFERENCES


143


REFERENCES


REFERENCES


Appendix A

Publications

A.1 Journal Papers


A.2 Conference Papers


